(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 874 299 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
28.10.1998 Bulletin 1998/44

(51) Int Cl.6: G06F 1/00, G06F 3/06, G11B 20/00

(21) Application number: 98303001.6

(22) Date of filing: 20.04.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 23.04.1997 JP 106104/97
02.06.1997 JP 143699/97
01.08.1997 JP 207953/97

(71) Applicant: SONY CORPORATION
Tokyo 141 (JP)

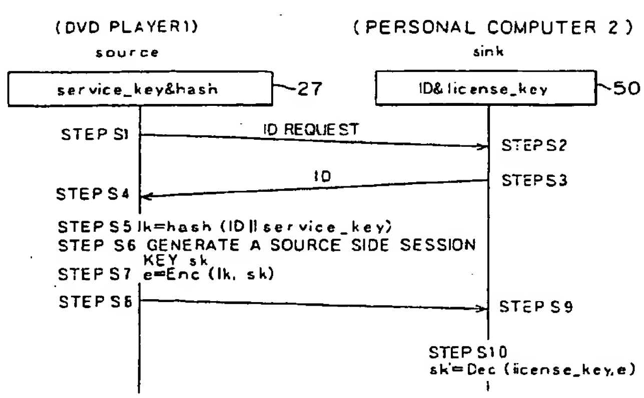
(72) Inventors:
- Ishiguro, Ryuji
Shinagawa-ku, Tokyo 141 (JP)
- Osawa, Yoshitomo
Shinagawa-ku, Tokyo 141 (JP)
- Osakabe, Yoshio
Shinagawa-ku, Tokyo 141 (JP)
- Sato, Makoto
Shinagawa-ku, Tokyo 141 (JP)
- Shima, Hisato
Shinagawa-ku, Tokyo 141 (JP)
- Asano, Tomoyuki
Shinagawa-ku, Tokyo 141 (JP)

(74) Representative: Pratt, Richard Wilson et al
D. Young & Co,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) Data transmission, reception, encryption, decryption and recording

(57) A hash function and a service key are stored in advance in an EEPROM of a DVD player serving as a source. In an EEPROM of a personal computer (PC) serving as a sink, on the other hand, its ID and a license key are stored beforehand. The DVD player requests the PC to transmit the ID. The DVD player then applies the hash function to data resulting from concatenation of the ID with the service key to generate a license key (= hash (ID|| service key)). Subsequently, the DVD player generates a source side session key and encrypts the session key by using the generated license key. Then, the DVD player transmits the encrypted source side session key to the PC. The PC decrypts the encrypted source side session key by using the license key stored in its EEPROM to produce a sink side session key which has a value equal to that of the source side session key.

FIG.4



EP 0 874 299 A2

Printed by Jouve, 75001 PARIS (FR)

BY EXPRESS MAIL NO. EL776051240US
Attorney Docket No. KOIK-T0343

Description

The present invention relates to a data transmission apparatus and method, a data reception apparatus and method, an encryption method and apparatus, a decryption apparatus and method and a recording medium.

Illustrative embodiments of the invention relate to such apparatus, methods and recording medium that allow data to be exchanged with a higher degree of security.

In recent years, there has been proposed a system comprising pieces of electronic equipment such as AV apparatuses and personal computers connected to each other by typically IEEE1394 serial buses wherein data can be exchanged among the pieces of equipment.

In such a system, for example, the ordinary user can play back movie information by using a DVD (Digital Video Disc) player and transmit the movie information to a monitor through the 1394 serial bus to display it on the monitor. The conduct done by the user to display the movie information is automatically permitted by the author of the movie information normally through a license which was obtained when the user purchased the DVD of the movie information. In order to do a conduct to copy the movie information played back from the DVD player to another recording medium such as an optical magnetic disc, however, it is necessary for the user to obtain a special permission from the author of the movie information. In the case of a copy license, typically, the optical magnetic disc apparatus is also used to store a key for indicating whether or not recording movie information into an optical magnetic disc mounted on the apparatus is allowed. That is to say, the key is used for forming a judgment as to whether or not the optical magnetic disc apparatus is a valid apparatus, that is, an apparatus licensed by the author of the movie information. If the optical magnetic disc apparatus is authenticated as a valid apparatus, the act to record the movie information into the apparatus can be judged to be a permitted conduct.

In such a case, it is necessary to verify that the destination apparatus is a valid apparatus in a transfer of information from an apparatus transmitting the information to an apparatus receiving the information, that is, the destination apparatus. It should be noted that the information transmitting apparatus and the information receiving apparatus are referred to hereafter as a source and a sink respectively.

Fig. 32 is a diagram showing the ordinary method for authenticating a destination apparatus. As shown in the figure, the source and the sink are each given a predetermined function f in advance by the author. Stored in a memory of each of the source and sink, the function f is difficult to identify from its input and output. In addition, it is difficult for a person who does not know the function f to infer an output produced by the function f from an input to the function f. The function f is provided to and stored in only an apparatus licensed by the author.

The source generates a random number r and transmits the number r to the sink through a 1394 serial bus. The source also applies the function f to the random number r, generating a number x (= f(r)).

Receiving the random number r from the source, the sink applies the function f to the random number r, generating a number y (= f(r)). The sink then transmits the number y to the source.

The source compares the calculated number x with the number y received from the sink to form a judgment as to whether or not the former is equal to the latter (x = y). If the number x is found equal to the number y, the source judges the sink to be a valid apparatus. In this case, movie information is encrypted by using a predetermined key before being transmitted to the sink.

As the key, a value k generated by applying the function f to the number y received by the source from the sink f is used (k = f(y)). By the same token, the sink also applies the function f to the number y to generate the value k (= f (y)). The value k is then, on the contrary, used as a key for decrypting the encrypted movie information.

In this method, however, it is necessary for all pieces of electronic equipment used as sources and sinks for transmitting and receiving information respectively to hold a uniform function f in strict confidence.

As a result, when the function f held in a piece of electronic is stolen by an unauthorized user, for example, the unauthorized user is capable of generating a key k by monitoring data exchanged by way of a 1394 serial bus and is, hence, capable of interpreting or decrypting encrypted data. In this way, the unauthorized user is capable of illegally stealing information by posing as an authorized user using a desired piece of electronic equipment.

Aspects of the invention are specified in the claims to which attention is invited.

Illustrative embodiments of the present invention seek to further improve security of transmitted information by preventing an unauthorized user from posing as a authorized user using a desired piece of electronic equipment even if data required for encrypting or decrypting the information is stolen by the unauthorized user.

The present invention will become more apparent and will hence be more readily appreciated as the same becomes better understood from a study of the following illustrative description of some preferred embodiments with reference to accompanying diagrams in which:

Fig. 1 is a block diagram showing a typical configuration of an information processing system to which an illustrative embodiment of the present invention is applied;

Fig. 2 is a block diagram showing detailed typical configurations of a DVD player 1, a personal computer 2 and an

optical magnetic disc apparatus 3 in the information processing system shown in Fig. 1;
Fig. 3 is an explanatory diagram used for describing authentication processing;
Fig. 4 is a diagram showing an embodiment implementing an authentication procedure for carrying out the authenticating processing shown in Fig. 3;
Fig. 5 is a diagram showing the format of a node unique ID;
Fig. 6 is a diagram showing another embodiment implementing the authentication procedure;
Fig. 7 is a diagram showing a further embodiment implementing the authentication procedure;
Fig. 8 is a diagram showing a still further embodiment implementing the authentication procedure;
Fig. 9 is a diagram showing still another embodiment implementing the authentication procedure;
Fig. 10 is a block diagram showing an embodiment implementing an information processing system to which an illustrative embodiment of the present invention is applied wherein a source transmits encrypted data to a plurality of sinks;
Fig. 11 is a block diagram showing a typical configuration of a 1394 interface unit 26 employed in a DVD player 1 serving as the source in the system shown in Fig. 10;
Fig. 12 is a block diagram showing a typical detailed configuration of the 1394 interface unit 26 shown in Fig. 11;
Fig. 13 is a block diagram showing a typical detailed configuration of an LFSR 72 employed in the 1394 interface unit 26 shown in Fig. 12;
Fig. 14 is a block diagram showing a more concrete configuration of the LFSR 72 shown in Fig. 13;
Fig. 15 is a block diagram showing a typical configuration of a 1394 interface unit 36 employed in an optical magnetic disc apparatus 3 serving as a sink in the system shown in Fig. 10;
Fig. 16 is a block diagram showing a typical detailed configuration of the 1394 interface unit 36 shown in Fig. 15;
Fig. 17 is a block diagram showing a typical configuration of a 1394 interface unit 49 employed in a personal computer 2 serving as another sink in the system shown in Fig. 10;
Fig. 18 is a block diagram showing a typical detailed configuration of the 1394 interface unit 49 shown in Fig. 17;
Fig. 19 is a block diagram showing a typical configuration of an application module 61 employed in the personal computer 2 serving as the other sink in the system shown in Fig. 10;
Fig. 20 is a block diagram showing a typical detailed configuration of the application module 61 shown in Fig. 19;
Fig. 21 is a block diagram showing another typical detailed configuration of the 1394 interface unit 26 employed in the DVD player 1 serving as the source in the system shown in Fig. 10;
Fig. 22 is a block diagram showing another typical detailed configuration of the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 serving as the sink in the system shown in Fig. 10;
Fig. 23 is a block diagram showing another typical detailed configuration of the 1394 interface unit 49 employed in the personal computer 2 serving as the other sink in the system shown in Fig. 10;
Fig. 24 is a block diagram showing another typical configuration of the application module 61 employed in the personal computer 2 serving as the other sink in the system shown in Fig. 10;
Fig. 25 is a diagram showing a still further embodiment implementing the authentication procedure;
Fig. 26 is a diagram showing a continuation procedure to the authentication procedure shown in Fig. 25;
Fig. 27 is a diagram showing an alternative continuation procedure to the authentication procedure shown in Fig. 25;
Fig. 28 is a block diagram showing the configuration of another embodiment implementing an information processing system to which an illustrative embodiment of the present invention is applied wherein a source transmits encrypted data to a sink;
Fig. 29 is a block diagram showing a random number generator 903 or 914 employed in the source or the sink respectively in the system shown in Fig. 28;
Fig. 30 shows a flowchart representing operations carried out by a processing circuit 902 or 913 employed in the source or the sink respectively in the system shown in Fig. 28;
Fig. 31 is a diagram showing a still further embodiment implementing the authentication procedure; and
Fig. 32 is a diagram showing the ordinary authentication procedure.

Fig. 1 is a block diagram showing a typical configuration of an information processing system to which an illustrative embodiment of the present invention is applied. As shown in the figure, in the configuration, a DVD player 1, a personal computer 2, an optical magnetic disc apparatus 3, a data broadcasting/receiving apparatus 4, a monitor 5 and a television receiver 6 are connected to each other by an IEEE1394 serial bus 11.

Fig. 2 is a block diagram showing detailed typical configurations of the DVD player 1, the personal computer 2 and the optical magnetic disc apparatus 3 in the information processing system shown in Fig. 1. The DVD player 1 comprises a CPU 21, a ROM unit 22, a RAM unit 23, an operation unit 24, a drive 25, a 1394 interface unit 26 and an EEPROM unit 27 which are connected to each other by an internal bus 28. As shown in the figure, the DVD player 1 is connected to the 1394 serial bus 11 through a 1394 interface unit 26. The CPU 21 carries out various kinds of processing by execution of a program stored in the ROM unit 22. The RAM unit 23 is used for properly storing information such as

data and the program which are required by the CPU 21 in carrying out the processing. The operation unit 24 comprises components such as buttons, switches and a remote controller. When the user operates the operation unit 24, a signal representing the operation is generated. The driver 25 drives a DVD which is not shown in the figure, playing back data recorded on the DVD. The EEPROM unit 27 is used for storing information which needs to be stored even after the power supply of the DVD player 1 is turned off. In the case of the present embodiment, an example of such information is an encryption /decryption key. The internal bus 28 is used for connecting the CPU 21, the ROM unit 22, the RAM unit 23, the operation unit 24, the drive 25, the 1394 interface unit 26 and the EEPROM unit 27 to each other.

Much like the DVD player 1, the optical magnetic disc apparatus 3 comprises a CPU 31, a ROM unit 32, a RAM unit 33, an operation unit 34, a drive 35, a 1394 interface unit 36 and an EEPROM unit 37 which are connected to each other by an internal bus 38. Since the CPU 31 to the internal bus 38 have the same functions of the CPU 21 to the internal bus 28 employed in the DVD player 1 respectively, their explanation is not repeated. The only exception is that the driver 35 drives an optical magnetic disc which is not shown in the figure instead of a DVD. The driver 35 records and plays back data into and from the optical magnetic disc.

In addition to a CPU 41, a ROM unit 42, a RAM unit 43, a 1394 interface unit 49 and an EEPROM unit 50 which are connected to each other by an internal bus 51, the personal computer 2 also includes an input/output interface unit 44, a keyboard 45, a mouse 46, an HDD (Hard Disc Drive) 47 and an expansion board 48. The personal computer 2 is connected to the 1394 serial bus 11 through the 1394 interface unit 49. The CPU 41 carries out various kinds of processing by execution of a program stored in the ROM unit 42. The RAM unit 43 is used for properly storing information such as data and the program which are required by the CPU 41 in carrying out the processing. Connected to the internal bus 51, the input/output interface unit 44 serves as an interface between the CPU 41 and the keyboard 45, the mouse 46, the HDD 47 and the expansion board 48. The input/output interface unit 44 passes on signals input from the keyboard 45 and the mouse 46 connected to the interface unit 44 to the CPU 41 by way of the internal bus 51. Connected to the HDD 47, the input/output interface unit 44 allows data and a program coming from the internal bus 51 to be stored into the HDD 47 and, on the contrary, data and a program stored in the HDD 47 to be read out and forwarded to the internal bus 51. The expansion board 48 is connected to the input/output interface unit 44, if needed, allowing necessary functions to be added to the personal computer 2. The EEPROM unit 50 is used for storing information which needs to be stored even after the power supply of the personal computer 2 is turned off. In the case of the present embodiment, an example of such information is a variety of encryption/decryption keys. The internal bus 51 is a local bus typically implemented by a PCI (Peripheral Component Interconnect) bus for connecting the CPU 41, the ROM unit 42, the RAM unit 43, the 1394 interface unit 49, the EEPROM unit 50 and the input/output interface unit 44 to each other.

It should be noted that the internal bus 51 is designed in an architecture open to the user through the input/output interface unit 44. That is to say, the user is allowed to connect an additional board as an expansion board 48 to the input/output interface unit 44, if required, and to write a custom program for the additional board to be installed in the personal computer 2. The CPU 41 then executes the custom program, properly exchanging data with the expansion board 48 by way of the internal bus 51 in order to implement a desired function.

In the case of a consumer electronic (CE) apparatus such as the DVD player 1 and the optical magnetic disc apparatus 3, on the contrary, their internal buses 28 and 38 are not designed in an architecture open to the user. Thus, the user is not capable of acquiring data transmitted by way of the internal bus 28 or 38 unless the internal bus 28 or 38 is redesigned specially.

The following is a description of processing of authentication of a sink carried out by a source with reference to Figs. 3 and 4. Fig. 3 is an explanatory diagram used for describing the authentication processing. As shown in the figure, the processing is typically carried out by firmware 20 stored as a program in advance in the ROM unit 22 employed in the DVD player 1 serving as the source to authenticate a license manager 62 stored in the ROM unit 42 to be executed as a program by the CPU 41 employed in the personal computer 2 serving as the sink.

Fig. 4 is a diagram showing an embodiment implementing a procedure whereby the source implemented typically by the DVD player 1 authenticates the sink implemented typically by the personal computer 2 by allowing the sink to generate a sink side session key having the same value as a source side session key generated by the source only if the sink is a valid sink. In the EEPROM unit 27 employed in the DVD player 1, a service key and a hash function are stored in advance. The service key and the hash function are given by an author of information to the user of the DVD player 1 who has to keep them in the EEPROM unit 27 in strict confidence.

The author provides the user with a service key for each piece of information created by the author. The service key is used as a key common to all apparatuses connected to each other by the 1394 serial bus 11 to compose a system. It should be noted that, in the present specification, the term system is used to imply the whole system comprising a plurality of apparatuses.

The hash function is used for transforming an input with an arbitrary length into output data with a fixed length such as 64 bits or 128 bits. Let the transformation be expressed by y = hash (x) where the symbol x is the input to the hash function and the symbol y is the data output by the function. In this case, the hash function is such a complex

function that it is difficult to find the value of x from a given value of y. The hash function is such a complicated function that it is difficult to find a pair of xl and x2 that satisfies the equation hash (xl) = hash (x2). MD5 and SHA are each the name of a function known as a representative one-way hash function. For details of the one-way hash function, refer to a reference with a title "Applied Cryptography" authored by Bruce Schneier, a second edition published by Wiley.

In the personal computer 2 used as a typical sink in the example shown in Fig. 4, on the other hand, an ID unique to the electronic apparatus, that is, the personal computer 2 in this case, and a license key provided in advance by the author of information are stored in strict confidence in the EEPROM unit 50. This node (apparatus) unique ID is normally assigned to the electronic apparatus by the manufacturer of electronic equipment as will be described later. The license key is a value resulting from application of the hash function to (n + m)-bit data which is obtained by concatenating the n-bit ID with the m-bit service key. Thus, the license key can be expressed by the following equation:

$$\text{license\_key} = \text{hash}(\text{ID} \,\|\, \text{service\_key})$$

where the notation "ID || service_key" represents a concatenation of the ID with the service key.

A node_unique_ID determined by specifications of the 1394 bus 11 can be typically used as an ID. Fig. 5 is a diagram showing the format of the node unique ID. As shown in the figure, the node_unique_ID comprises 8 bytes (or 64 bits). The first 3 bytes are controlled by the IEEE and given by the IEEE to a manufacturer of electronic equipment as a number unique to the manufacturer. On the other hand, the low-order 5 bytes can be assigned by the manufacturer of electronic equipment itself to an electronic apparatus sold to the user. Typically, each value of the whole low-order 5 bytes are assigned by the electronic equipment maker to an electronic apparatus as a serial number of the apparatus. Since the high-order 3 bytes have a value unique to the manufacturer of electronic equipment, the node_unique_ID is unique to each of electronic apparatuses without regard to whether the apparatuses are produced by the same manufacturer or different manufacturers.

As shown in Fig. 4, the procedure begins with a step S1 at which the firmware 20 in the DVD player 1 controls the 1394 interface unit 26 to make a request to the personal computer 2 for the ID thereof to be transmitted by way of the 1394 serial bus 11. Then, the procedure goes on to a step S2 at which the license manager 62 of the personal computer 2 receives the request for the ID. To put it in detail, the 1394 interface unit 49 employed in the personal computer 2 passes on the request for the ID transmitted by the DVD player 1 by way of the 1394 serial bus 11 to the CPU 41. The procedure then proceeds to a step S3 at which the license manager 62 being executed by the CPU 41 reads out the ID from the EEPROM unit 50 in accordance with the request forwarded thereto by the 1394 interface unit 49 and transmits the ID to the DVD player 1 by way of the 1394 interface unit 49 and the 1394 serial bus 11.

Then, the procedure continues to a step S4 at which the 1394 interface unit 26 employed in the DVD player 1 receives the ID and passes on it to the firmware 20 being executed by the CPU 21.

Subsequently, the procedure goes on to a step S5 at which the firmware 20 concatenates the ID received from the personal computer 2 with a service key stored in the EEPROM unit 27 to form data (ID || service_key). Then, a license key lk is computed by applying the hash function to the data (ID || service_key) as shown in the following equation:

$$lk = \text{hash}(\text{ID} \,\|\, \text{service\_key})$$

The procedure then proceeds to a step S6 at which the firmware 20 generates a source side session key sk, details of which will be described later. The source side session key sk will be used as a common session key S by both the DVD player 1 to encrypt a clear text to be transmitted and by the personal computer 2 to decrypt an encrypted text received from the DVD player 1.

Then, the procedure continues to a step S7 at which the firmware 20 encrypts the source side session key sk generated at the step S6 by using the license key lk computed at the step S5 as a key to produce an encrypted source side session key e in accordance with the following equation:

$$e = \text{Enc}(lk, sk)$$

It should be noted that the expression Enc (A, B) on the right hand side of the above equation represents a common session key encryption / decryption technique whereby data B is encrypted by using a key A to produce an encrypted source side session key e on the left hand side of the equation.

Subsequently, the procedure goes on to a step S8 at which the firmware 20 transmits the encrypted source side session key e generated at the step S7 to the personal computer 2. To put it in detail, the encrypted source side session

key e is transmitted by the 1394 interface unit 26 employed in the DVD player 1 to the personal computer 2 by way of the 1394 serial bus 11. The procedure then proceeds to a step S9 at which the 1394 interface unit 49 employed in the personal computer 2 receives the encrypted source side session key e. Then, the procedure proceeds to a step S10 at which the license manager 62 decrypts the encrypted source side session key e passed on thereto by the 1394 interface unit 49 by using a license key provided in advance by the author of information and stored in the EEPROM unit 50 as a key to produce a sink side session key sk' in accordance with the following equation:

$$sk' = Dec\ (license\_key,\ e)$$

It should be noted that the expression Dec (A, B) on the right hand side of the above equation represents the common session key encryption/decryption technique whereby encrypted data B is in this case decrypted by using a key A to produce a sink side session key sk' on the left hand side of the equation.

It is also worth noting that a DES algorithm is known as a data encrypting/decrypting algorithm adopted in the common session key encryption/decryption technique which is also described in detail in the second edition of the reference with the title "Applied Cryptography" cited above.

The license key provided by the author of information and stored in the EEPROM unit 50 employed in the personal computer 2 in advance has a value which was computed by the author by using the same hash function as license the key lk was generated by the DVD player 1 at the step S5. That is to say, the following equation holds true:

$$lk = license\_key$$

Thus, based on the common source side session key encryption/decryption technique using the same (license) key, the decryption carried out by the personal computer 2 at the step S10 is just a reversed process of the encryption performed by the DVD player 1 at the step S7. As a result, since e is the encrypted data of the source side session key sk generated by the DVD player 1 at the step S6, the sink side session key sk' computed by the personal computer 2, that is, a result of the decryption of the encrypted source side session key e, is equal to the source side session key sk. That is to say, the following equation holds true:

$$sk' = sk$$

In this way, since the source and sink side session keys sk and sk' have the same value, the source implemented typically by the DVD player 1 and the sink implemented typically by the personal computer 2 can share a common session key S. For this reason, the DVD player 1 can use the key sk as an encryption key as it is to encrypt a clear text created by the author to be transmitted to the personal computer 2. By the same token, the personal computer 2 can use the sink side session key sk' as a decryption key as it is to decrypt an encrypted text received from the DVD player 1. As an alternative, the DVD player 1 generates a pseudo random number to be used as an encryption key by using the source side session key sk as a base as will be described later. Likewise, the personal computer 2 generates a random number to be used as a decryption key by using the sink side session key sk' as a base as will also be described later.

As described above, the license key lk is generated at the step S5 of the procedure shown in Fig. 4 by applying the hash function to a concatenation of an ID unique to a particular electronic apparatus and a service key provided for a text created by the author. Thus, in a pair of electronic apparatuses wherein the source does not have the service key for the text and/or the sink does not have the ID unique to the legal owner, it is impossible to generate the correct license key lk (Refer to the step S5 of the procedure shown in Fig. 4). In addition, an electronic apparatus not authenticated by the author is not provided with a license key and, thus, not capable of generating the session key sk' (Refer to the step S10 of the procedure shown in Fig. 4). In a normal case, after the procedure shown in Fig. 4 is completed, the DVD player 1 encrypts reproduced data or a clear text by using the source side session key sk and transmits the encrypted data or the encrypted text to the personal computer 2. Provided with a correct license key, the personal computer 2 is capable of generating the sink side session key sk' (Refer to the step S10 of the procedure shown in Fig. 4). The personal computer 2 is thus capable of decrypting the encrypted playback data or the encrypted text received from the DVD player 1 by means of the sink side session key sk'. If the personal computer 2 is not a licensed electronic apparatus, however, it will be impossible to generate the sink side session key sk' because the correct license key is not available. As a result, the unlicensed personal computer 2 is not capable of decrypting the encrypted playback data or the encrypted text received from the DVD player 1. In other words, only a sink capable of generating a sink side session key sk' having the same value as the source side session key sk generated by the source is authenticated

in the end. This is because only a particular electronic apparatus serving as an authorized source which has a service key provided by an author for information or a text created by the author and receives a correct ID from an authorized sink is capable of generating the correct license key lk. By the same token, only a particular electronic apparatus serving as an authorized sink which is provided with the correct license key by the author is capable of generating the correct sink side session key sk' for use as a decryption key to decrypt encrypted data or an encrypted text.

Assume that a license key granted to a personal computer 2 is stolen by any chance. In this case, nevertheless, the stolen license key can not be used in another electronic apparatus to generate a valid sink side session key sk' because the other apparatus has an ID different from that assigned to the personal computer 2. Since the ID varies from apparatus to apparatus as such, another electronic apparatus will not be capable of decrypting the encrypted playback data or the encrypted text received from the DVD player 1 by means of the stolen license key. As a result, the security of transmitted information can be enhanced.

Fig. 6 is a diagram showing another embodiment implementing an authentication procedure whereby a source implemented typically by the DVD player 1 authenticates two sinks implemented typically by the personal computer 2 and the optical magnetic disc apparatus 3 respectively by allowing each of the sinks to generate a sink side session key having the same value as a source side session key generated by the source only if the sinks are valid sinks.

In the EEPROM unit 50 employed in the personal computer 2 serving as the first sink, ID 1, an identification assigned in advance uniquely by a manufacturer of electronic equipment to the personal computer 2, and License Key 1, a license key provided in advance by an author of information to the computer 2 are stored. By the same token, in the EEPROM unit 37 employed in the optical magnetic disc apparatus 3 serving as the second sink, ID 2, an ID assigned in advance uniquely by a manufacturer of electronic equipment to the disc apparatus 3, and License Key 2, a license key provided in advance by the author of information to the disc apparatus 3 are stored.

Since pieces of processing carried out at the steps S11 to S20 by the DVD player 1 serving as the source and the personal computer 2 serving as the first sink are in essence the same as those of the steps S1 to S10 of the procedure shown in Fig. 4, their explanation is not repeated.

In brief, the personal computer 2 generates a valid sink side session key sk1' from an encrypted source side session key e1 received from the DVD player 1 at the step S20 as described above. The procedure then goes on to a step S21 at which the firmware 20 in the DVD player 1 controls the 1394 interface unit 26 to make a request to the optical magnetic disc apparatus 3 for the ID thereof to be transmitted by way of the 1394 serial bus 11. Then, the procedure goes on to a step S22 at which firmware 30 of the optical magnetic disc apparatus 3 shown in Fig. 10 receives the request for the ID. To put it in detail, the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 passes on the request for the ID transmitted by the DVD player 1 by way of the 1394 serial bus 11 to the CPU 31. The procedure then proceeds to a step S23 at which the firmware being executed by the CPU 31 reads out the identification ID2 from the EEPROM unit 37 in accordance with the request forwarded thereto by the 1394 interface unit 36 and transmits the identification ID2 to the DVD player 1 by way of the 1394 interface unit 36 and the 1394 serial bus 11.

Then, the procedure continues to a step S24 at which the 1394 interface unit 26 employed in the DVD player 1 receives the identification ID2 and passes on it to the firmware 20 being executed by the CPU 21.

Subsequently, the procedure goes on to a step S25 at which the firmware 20 concatenates the identification ID2 received from the optical magnetic disc apparatus 3 with a service key stored in the EEPROM unit 27 to form data (ID2 || service_key). Then, a license key lk2 is computed by applying the hash function to the data (ID2 || service_key) as shown in the following equation:

$$lk2 = \mathrm{hash}\ (ID2\ \|\ service\_key)$$

Then, the procedure continues to a step S26 at which the firmware 20 encrypts the source side session key sk generated at the step S16 by using the license key lk2 computed at the step S25 as a key to produce an encrypted source side session key e2 in accordance with the following equation:

$$e2 = \mathrm{Enc}\ (lk2,\ sk)$$

Subsequently, the procedure goes on to a step S27 at which the firmware 20 transmits the encrypted source side session key e2 generated at the step S26 to the optical magnetic disc 3. To put it in detail, the encrypted source side session key e2 is transmitted by the 1394 interface unit 26 employed in the DVD player 1 to the optical magnetic disc apparatus 3 by way of the 1394 serial bus 11.

The procedure then proceeds to a step S28 at which the 1394 interface unit 36 employed in the optical magnetic disc 3 receives the encrypted source side session key e2. Then, the procedure proceeds to a step S29 at which the

firmware 30 decrypts the encrypted source side session key e2 passed on thereto by the 1394 interface unit 36 by using a license key (license_key 2) stored in the EEPROM unit 37 as a key to produce a sink side session key sk2' in accordance with the following equation:

$$sk2' = \mathrm{Dec}\ (\mathrm{license\_key}\ 2,\ e2)$$

As described above, the personal computer 2 and the optical magnetic disc apparatus 3 generate the sink side session keys sk1' and sk2' at the steps S20 and S29 respectively. Normally, the sink side session keys sk1' and sk2' have the same value as the source side session key sk generated by the DVD player 1 at the step S16.

In the procedure shown in Fig. 6, the DVD player 1 makes requests for an ID to the personal computer 2 and the optical magnetic disc apparatus 3 separately. It should be noted, however, that in the case of broadcasting communication wherein requests can be made at the same time, processing according to an embodiment implementing a procedure like one shown in Fig. 7 can be carried out.

As shown in the figure, the procedure begins with a step S41 at which the DVD player 1 transmits requests to all sinks, that is, the personal computer 2 and the optical magnetic disc apparatus 3, for the IDs thereof by broadcasting communication. Then, the procedure goes on to steps S42 and S43 at which the personal computer 2 and the optical magnetic disc apparatus 3 respectively receive the requests for the IDs. The procedure then proceeds to steps S44 and S45 at which the personal computer 2 and the optical magnetic disc apparatus 3 read out the identifications ID1 and ID2 from the EEPROM units 50 and 37 respectively and transmit them to the DVD player 1. Then, the procedure continues to steps S46 and S47 at which the DVD player 1 receives the identifications ID1 and ID2 respectively.

Subsequently, the procedure goes on to a step S48 at which the DVD player 1 concatenates the identification ID1 received from the personal computer 2 with a service key stored in the EEPROM unit 27 to form data (ID1 || service_key). Then, a license key lk1 is computed by applying the hash function to the data (ID1 || service_key) as shown in the following equation:

$$lk1 = \mathrm{hash}\ (ID1\ \|\ \mathrm{service\_key})$$

Subsequently, the procedure goes on to a step S49 at which the DVD player 1 concatenates the identification ID2 received from the optical magnetic disc apparatus 3 with the service key stored in the EEPROM unit 27 to form data (ID2 || service_key). Then, a license key lk2 is computed by applying the hash function to the data (ID2 || service_key) as shown in the following equation:

$$lk2 = \mathrm{hash}\ (ID2\ \|\ \mathrm{service\_key})$$

The procedure then proceeds to a step S50 at which the DVD player 1 generates a source side session key sk. Then, the procedure continues to a step S51 at which the DVD player 1 encrypts the source side session key sk generated at the step S50 by using the license key lk1 computed at the step S48 as a key to produce an encrypted source side session key e1 in accordance with the following equation:

$$e1 = \mathrm{Enc}\ (lk1,\ sk)$$

Then, the procedure continues to a step S52 at which the DVD player 1 encrypts the source side session key sk generated at the step S50 by using the license key lk2 computed at the step S49 as a key to produce an encrypted source side session key e2 in accordance with the following equation:

$$e2 = \mathrm{Enc}\ (lk2,\ sk)$$

The procedure then goes on to a step S53 at which the identification ID1, the encrypted source side session key e1, the identification ID2 and the encrypted source side session key e2 are concatenated to produce encrypted data e as follows:

$$e = ID1\ \|\ e1\ \|\ ID2\ \|\ e2$$

Subsequently, the procedure goes on to a step S54 at which the DVD player 1 transmits the encrypted data e to the personal computer 2 and the optical magnetic disc apparatus 3 by broadcasting communication. The procedure then proceeds to steps S55 and S56 at which the personal computer 2 and the optical magnetic disc apparatus 3 receive the encrypted data e. Then, the procedure proceeds to steps S57 and S58 at which the personal computer 2 and the optical magnetic disc apparatus 3 decrypt the encrypted source side session keys e1 and e2 extracted from the encrypted data e by using the license keys License Key 1 and License Key 2 stored in the EEPROM units 50 and 37 as keys to produce sink side session keys sk1' and sk2' respectively in accordance with the following equations:

$$sk1' = Dec\ (License\_Key\ 1,\ e1)$$

$$sk2' = Dec\ (License\_Key\ 2,\ e2)$$

Fig. 8 is a diagram showing an embodiment implementing a procedure of authentication processing whereby only a valid sink will generate a sink side session key sk' having the same value as a source side session key sk generated by a source in a system wherein the sink is capable of rendering a plurality of services, that is, decrypting a plurality of kinds of information. To handle the different kinds of information, the personal computer 2 serving as the sink is provided with a plurality of license keys stored in the EEPROM unit 50 such as License_Key 1, License_Key 2, License_Key 3 etc. for the different kinds of information. By the same token, the DVD player 1 serving as a source has information on a plurality of service IDs for identifying which kinds of information to be transmitted to the sink and a plurality of service keys stored in the EEPROM unit 27 such as Service_Key 1, Service_Key 2, Service_Key 3 etc. used for generating License_Key 1, License_Key 2, License_Key 3 etc. respectively. Pieces of processing carried out in the procedure shown in Fig. 8 are similar to those of the procedure shown in Fig. 4 except for the following steps. To begin with, at a step S81, the DVD player 1 transmits a request for an ID along with a service ID for identifying a kind of information, which is to be serviced by the personal computer 2 used as the sink, to the personal computer 2. Then, at a step S85, a license key lk is generated by the DVD player 1 by application of the hash function to an ID received from the personal computer 2 and one of Service_Key 1, Service_Key 2, Service_Key 3 etc. in the EEPROM unit 27 which is associated with the kind of information to be transmitted to the sink, that is, associated with the service ID transmitted to the personal computer 2 at the step S81. Finally, at a step S90, the personal computer 2 generates a sink side session key sk' from an encrypted source side session key e received from the DVD player 1 at a step 89 and one of License_Key 1, License_Key 2, License_Key 3 etc. in the EEPROM unit 50 that is associated with the service ID received from the DVD player 1 at the step S82.

Fig. 9 is a diagram showing another embodiment implementing a procedure of authentication whereby only a valid sink will be capable of generating a sink side session key sk' having the same value as a source side session key sk generated by a source. In this case, the DVD player 1 used as a source has a service key, a hash function and a pseudo random number generating function pRNG which are stored in the EEPROM unit 27 employed thereby. The service key, the hash function and the pseudo random number generating function pRNG are given by an author of information and kept in strict confidence. On the other hand, stored in the EEPROM unit 50 employed by the personal computer 2 serving as a sink are an ID assigned to the personal computer 2 by the manufacturer of electronic equipment as well as license keys LK and LK', a confusion function G and the pseudo random number generating function pRNG which are given by the author of the information.

The license key LK is a unique random number generated by the author whereas the license key LK' is also generated by the author so as to satisfy the following equation:

$$LK' = G\ \wedge - 1\ (R)$$

where R = pRNG (H) (+) pRNG (LK)
where H = hash (ID || service_key)

It should be noted that, while the symbol ^ alone denotes the power notation, the notation "G ^ -1" means the inverse function of the confusion function G. The value of the inverse function G ^ -1 can be found with ease provided that predetermined rules are known. If the predetermined rules are not known, however, it is difficult to compute the value of the inverse function G ^ -1. A function used in encryption based on a disclosed key can be utilized as this function.

In addition, the function pRNG for generating a random number can be implemented by hardware.

As shown in Fig. 9, the procedure begins with a step S101 at which the firmware20 in the DVD player 1 makes a request to the license manager 62 of the personal computer 2 for the ID thereof to be transmitted. Then, the procedure

goes on to a step S102 at which the license manager 62 of the personal computer 2 receives the request for the ID. The procedure then proceeds to a step S103 at which the license manager 62 reads out the ID from the EEPROM unit 50 in accordance with the request and transmits the ID to the DVD player 1. Then, the procedure continues to a step S104 at which the DVD player 1 receives the ID. Subsequently, the procedure goes on to a step S105 at which the firmware 20 concatenates the ID received from the personal computer 2 with a service key stored in the EEPROM unit 27 to form data (ID || service_key). Then, a value H is computed by applying the hash function to the data (ID || service_key) as shown in the following equation:

$$H = \text{hash}\ (ID \,\|\, \text{service\_key})$$

The procedure then proceeds to a step S106 at which the firmware 20 generates a source side session key sk. Then, the procedure continues to a step S107 at which the firmware 20 compute an encrypted source side session key e from the value H generated at the step S105 and the source side session key sk generated at the step S106 in accordance with the following equation:

$$e = sk\ (+)\ pRNG\ (H)$$

where the notation (+) used on the right hand side of the above equation is the operator of the operation to compute an exclusive logical sum and, thus, an expression A (+) B represents the exclusive logical sum of A and B.

That is to say, at the step S107, the source side session key sk generated at the step S106 is encrypted to produce the encrypted source side session key e by finding the exclusive logical sum of each bit of the key sk and the corresponding bit of pRNG (H), a random number obtained by applying the pseudo random number generating function pRNG to the value H generated at the step S105.

Subsequently, the procedure goes on to a step S108 at which the firmware20 transmits the encrypted source side session key e generated at the step S107 to the personal computer 2.

The procedure then proceeds to a step S109 at which the personal computer 2 receives the encrypted source side session key e. Then, the procedure proceeds to a step S110 at which the license manager 62 decrypts the encrypted source side session key e by using the license keys LK and LK' stored in the EEPROM unit 50 as keys to produce a sink side session key sk' in accordance with the following equation:

$$sk' = e\ (+)\ G\ (LK')\ (+)\ pRNG\ (LK)$$

That is to say, at the step S110, the encrypted source side session key e received from the DVD player 1 is decrypted to produce the sink side session key sk' by finding the exclusive logical sum of the encrypted source side session key e , G (LK'), a value obtained by applying the confusion function G stored in the EEPROM unit 50 to the license key LK' also stored in the EEPROM unit 50, and pRNG (LK), a value obtained by applying the pseudo random number generating function pRNG also stored in the EEPROM unit 50 to the license key LK also stored in the EEPROM unit 50.

Much like the procedure shown in Fig. 4, the sink side session key sk' generated by the personal computer 2 at the step S110 has the same value as the source side session key sk generated by the DVD player 1 at the step S6. The fact that sk = sk' is proven by the following:

$$sk' = e\ (+)\ G\ (LK')\ (+)\ pRNG\ (LK)$$

Substituting (sk (+) pRNG (H)) for e in the expression on the right hand side of the above equation yields the following equation:

$$sk' = sk\ (+)\ pRNG\ (H)\ (+)\ G(LK')\ (+)\ pRNG\ (LK)$$

Since G(LK') = G(G ^ - 1 (R)) = R, the following equation is obtained:

$$sk' = sk\ (+)\ pRNG\ (H)\ (+)\ R\ (+)\ pRNG\ (LK)$$

Substituting (pRNG (H) (+) pRNG (LK)) for R in the expression on the right hand side of the above equation yields the following equation:

$$sk' = sk \ (+) \ pRNG \ (H) \ (+) \ pRNG \ (H) \ (+) \ pRNG \ (LK) \ (+) \ pRNG \ (LK)$$
$$= sk$$

As described above, the source and sink side session keys sk and sk' are a common key S shared by both the DVD player 1 and the personal computer 2 serving as a source and a sink respectively. In addition, unlike the procedures described previously, it is only an author of information who is capable of generating license keys LK and LK'. Thus, an attempt made by a source to illegally generate the license keys LK and LK' will end in a failure. As a result, the security of transmitted information can be further improved.

In the authentication procedures described above, a source authenticates a sink by allowing the sink to generate a sink side session key sk' having the same value as a source side session key sk generated by the source only if the sink is a valid sink. The procedure can also be applied for example to authenticate the ordinary operation to load an application program in the personal computer 2 in order to prevent an application program obtained illegally from being executed. In this case, it is necessary to form a judgment as to whether or not execution of each application program is allowed by the author of the program through the same procedure as those described so far whereby the license manager 62 authenticates an application module 61 as shown in Fig. 3. To be more specific, in the authentication procedure shown in Fig. 3, the license manager 62 serves as a source whereas the application module 61 is used as a sink.

After the authentication process described above has been completed, that is, after the sink has generated a sink side session key sk' having the same value as a source side session key sk generated by the source, data or a clear text encrypted by the source by using an encryption key is transmitted to the sink from the source. At the sink, the encrypted data or the encrypted text is decrypted back by using a decryption key. As described above, the source and sink side session keys sk and sk' can be used as encryption and decryption keys respectively as they are or, as an alternative, a random number generated from the session key sk or sk' is used as an encryption or decryption key instead. The operation carried out by the source to encrypt data and the operation carried out by the sink to decrypt the encrypted data are explained as follows.

In an electronic apparatus such as the DVD player 1 and the optical magnetic disc apparatus 3, the internal functions of which are not built in an architecture open to the user, the processing to encrypt and decrypt data transmitted through the 1394 serial bus 11 in a system like one shown in Fig. 10, a block diagram showing a system wherein a source transmits encrypted data to sinks, is carried out by the 1394 interface units 26 and 36 employed in the DVD player 1 and the optical magnetic disc apparatus 3 respectively. Data is encrypted or decrypted by using a session key S, that is, the source side session key sk or the sink side session key sk' described earlier, and a time variable key i, strictly speaking, a key i' for generating the time variable key i. The session key S and the key i' are supplied by the firmware20 or 30 to the 1394 interface unit 26 or 36 respectively. The session key S comprises an initial value key Ss used as an initial value and a derangement key Si for deranging the time variable key i. The initial value key Ss and the derangement key Si can be formed respectively from a predetermined number of high order bits and a predetermined number of low order bits of the source side session key sk or the sink side session key sk' which has the same value as sk used in the process of authenticating the sink described earlier. The session key S is properly updated in each session, for example, for each movie information or for each playback operation. On the other hand, the time variable key i which is generated from the derangement key Si of the session key S and the key i' is updated a number of times in a session. For example, time information obtained with predetermined timing can be used typically as the key i'.

Assume that movie data played back and output by the DVD player 1 serving as a source is transmitted to the optical magnetic disc apparatus 3 and the personal computer 2 which are used as sinks by way of the 1394 serial bus 11 and is then decrypted by the sinks. In this case, the data is encrypted by the 1394 interface unit 26 employed in the DVD player 1 by using the session key S and the time variable key i, strictly speaking, the key i' and the encrypted data is decrypted back by the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 by using the session key S and the time variable key i, strictly speaking, the key i'.

In the personal computer 2, on the other hand, the license manager 62 supplies the initial value key Ss of the session key S to the application module 61 and the derangement key Si of the session key S and the time variable key i, strictly speaking, the key i' for generating the time variable key i, to the 1394 interface unit 49 serving as a link unit. In the 1394 interface unit 49, the time variable key i is generated from the derangement key Si and the key i' and

used for decrypting back the encrypted data. The decrypted data is further decrypted by the application module 61 by using the session key S, strictly speaking, by using the initial value key Ss of the session key S.

As described above, in the personal computer 2 having an architecture wherein the internal bus 51 is designed in an architecture open to the user, the 1394 interface unit 49 carries out only a 1st stage of the decryption on the encrypted data, leaving the data still in an encrypted state. Then, the application module 61 further performs a 2nd stage of the decryption on the data decrypted by the 1394 interface unit 49 to produce the clear text. In this way, the personal computer 2 is prohibited from copying data (that is, a clear text) transferred by way of the internal bus 51 to another medium such as a hard disc mounted on the hard disc drive 47 through the use of a proper function added to the internal bus 51.

As described above, according to the embodiment of the present invention, in a CE apparatus with an architecture wherein an internal bus is not open to the user, encrypted data is decrypted only once by using a session key S and a time variable key i, strictly speaking, a key i'. In the case of a CE apparatus such as the personal computer 2 with an architecture wherein an internal bus is open to the user, on the other hand, encrypted data is decrypted by using a time variable key i, which is generated by using the derangement key Si of a session key S and the key i', at a 1st stage of decryption, and then further decrypted by using the initial value Ss of the session key S at a 2nd stage of decryption. Thelst and 2nd stages of the decryption processing are represented by the following equation:

$$\mathrm{Dec}\ (Ss,\ \mathrm{Dec}\ (i,\ \mathrm{Enc}\ (\mathrm{algo}\ (S + i'),\ \mathrm{Data}))) = \mathrm{Data}$$

where the term algo (S + i') appearing on the left hand side of the above equation represents a value resulting from application of a predetermined algorithm to the session key S and the time variable key i, strictly speaking, the key i', the notation Dec appearing at the left end of the equation represents the 2nd stage of the decryption, the other Dec notation denotes the 1st stage of decryption and the notation Enc indicates the encryption carried out by the source.

Fig. 11 is a block diagram showing a typical configuration of the 1394 interface unit 26 that satisfies the term Enc appearing in the equation given above to represent the encryption carried out by the DVD player 1 employing the 1394 interface unit 26. As shown in the figure, the configuration comprises an additive generator 71, an LFSR (Linear Feedback Shift Register) 72, a shrink generator 73 and an adder 74. m-bit data generated by the additive generator 71 and 1-bit data generated by the LFSR are supplied to the shrink generator 73. The shrink generator 73 selects some pieces of m-bit data received from the additive generator 71 in accordance with the value of the 1-bit data supplied by the LFSR 72 and outputs the selected m-bit data to the adder 74 as an encryption key. It should be noted that the m-bit encryption key, a random number generated by the shrink generator 73, corresponds the key (S + i') in the equation given above. The adder 74 adds the m-bit encryption key received from the shrink generator 73 to an input clear text, that is, m-bit data to be transmitted to the 1394 serial bus 11, to produce an encrypted text or encrypted data.

The addition carried out by the adder 74 is a mod 2 ^ m process, where the symbol ^ is the power notation, meaning addition of the encryption key generated by the shrink generator 73 to the clear text. In other words, the process is addition of an m-bit key to m-bit data with a carry-over ignored.

Fig. 12 is a block diagram showing a detailed configuration of the 1394 interface unit 26 which is shown in Fig. 11 in a simple and plain manner. As shown in Fig. 12, the initial value key Ss of the session key S received from the firmware20 is supplied to and held in a register 82 by way of the adder 81. Typically, the initial value key Ss comprises 55 words each having a length in the range 8 to 32 bits. On the other hand, the derangement key Si of the session key S is held in a register 85. Typically, the derangement key Si is the low order 32 bits of the session key S.

The key i' is held in a 32-bit register 84. The key i' is created in a process of accumulation of bits. To put it in detail, each time a packet is transmitted through the 1394 serial bus 11, typically, two bits used for forming the key i' are supplied to the register 84. The creation of the 32-bit key i' is completed as 16 packets are transmitted. At that time, the 32-bit key i' is added to the derangement key Si held in the register 85 by an adder 86 to finally generate a time variable key i which is supplied to the adder 81. The adder 81 adds the time variable key i output by the adder 86 to the initial value key Ss held in the register 82, storing the result of the addition back in the register 82.

Assume that the number of bits per word in the register 82 is 8. In this case, since the time variable key i output by the adder 86 is 32 bits in width, the time variable key i is divided into 4 portions each comprising 8 bits. Each of the 4 portions is then added to a word in the register 82 at a predetermined address, that is, at one of the addresses 0 to 54.

As described above, the initial value key Ss is held initially in the register 82. Each time16 packets of an encrypted text are transmitted thereafter, however, the initial value Ss is updated by adding the time variable key i thereto.

An adder 83 selects predetermined two words among the 55 words of the register 82 and adds the selected two words to each other. With timing shown in Fig. 12, words at addresses 23 and 54 are selected by the adder 83. The adder 83 supplies the result of the addition to the shrink generator 73 and a word in the register 82. With the timing shown in Fig. 12, the adder 83 supplies the result of the addition to the word of the register 82 at an address 0 to replace the data currently stored in the word.

At the next timing, the two words selected by the adder 83 are changed from the addresses 54 and 23 to addresses 53 and 22, being shifted in the upward direction shown in the figure by 1 word. By the same token, the destination of the result of the addition output by the adder 83 is also shifted upward. Since there is no word above address 0, however, the destination is changed from the word at address 0 to the word at address 54 at the bottom of the register 82.

It should be noted that, in each of the adders 81, 83 and 86, processing to compute an exclusive logical sum can be carried out instead.

Fig. 13 is a block diagram showing a typical configuration of the LFSR 72. As shown in the figure, the LFSR 72 comprises an n-bit shift register 101 and an adder 102 for summing up the values of a predetermined number of bits among the n bits. A bit resulting from the addition by the adder 102 is stored in the left most bit b_n of the n-bit shift register 101 shown in the figure and, at the same time, the previous value of the bit b_n is shifted to a bit b_{n-1} on the right hand side of the bit bn. By the same token, the bit shifting to the right is applied to the previous values of bits b_{n-1}, b_{n-2}, ---, etc. whereas the previous value of the right most bit b_1 shown in the figure is output. At the next timing, a bit resulting from the addition by the adder 102 is again stored in the left most bit b_n of the n-bit shift register 101 and, at the same time, the previous value of the bit bn is again shifted to a bit b_{n-1} on the right hand side of the bit b_n. By the same token, the bit shifting to the right is gain applied to the previous values of bits b_{n-1}, b_{n-2}, ---, etc. whereas the previous value of the right most bit b_1 is again output. These operations are carried out repeatedly, sequentially outputting bits from the right most bit b_1 one bit after another.

Fig. 13 is a diagram showing a typical configuration of the LFSR 72 in general terms. On the other hand, Fig. 14 is a diagram showing a typical configuration of the LFSR 72 in more concrete terms. In the configuration shown in Fig. 14, the shift register 101 comprises 31 bits. The adder 102 is used for adding the value of the left most bit b_{31} to the value of the right most bit b_1 and storing the result of the addition in the left most bit 31 of the shift register 101.

As shown in Fig. 12, the shrink generator 73 comprises a condition judging unit 91 and a FIFO unit 92. The condition judging unit 91 passes on m-bit data supplied by the adder 83 employed in the additive generator 71 to the FIFO unit 92 to be held therein as it is when the LFSR 72 outputs a bit having the logic value "1". When the LFSR 72 outputs a bit having the logic value "0", on the other hand, the condition judging unit 91 does not pass on m-bit data supplied by the adder 83 employed in the additive generator 71 to the FIFO unit 92, suspending the encryption process. In this way, the condition judging unit 91 employed in the shrink generator 73 selects only pieces of m-bit data which are each generated by the additive generator 71 while the LFSR 72 is outputting a bit with the logic value "1" and stores the selected piece of m-bit data in the FIFO unit 92 of the generator 73.

Each piece of m-bit data held in the FIFO unit 92 is supplied as an encryption key to the adder 74 for generating an encrypted text by adding the encryption key to data representing a clear text to be transmitted to a sink, that is, data played back from a DVD in the source.

The encrypted data is then transmitted from the DVD player 1 to the optical magnetic disc apparatus 3 and the personal computer 2 by way of the 1394 serial bus 11.

Fig. 15 is a diagram showing a typical configuration of the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 for decrypting the encrypted data received from the DVD player 1 by way of the 1394 serial bus 11. As shown in the figure, much like the 1394 interface unit 26 employed in the DVD player 1 shown in Fig.11, the configuration comprises an additive generator 171, an LFSR (Linear Feedback Shift Register) 172, a shrink generator 173 and a subtractor 174. m-bit data generated by the additive generator 171 and 1-bit data generated by the LFSR 172 are supplied to the shrink generator 173. The shrink generator 173 selects some pieces of m-bit data received from the additive generator 171 in accordance with the value of the 1-bit data supplied by the LFSR 172 and outputs the selected m-bit data to the subtractor 174 as a decryption key. The subtractor 174 subtracts the m-bit decryption key received from the shrink generator 173 from an encrypted text, that is, m-bit data received from the DVD player 1 by way of the 1394 serial bus 11, to decrypt the encrypted text back into the clear text.

It is obvious that the configuration of the 1394 interface unit 36 employed in the DVD player 1 shown in Fig. 15 is basically identical with that of the 1394 interface unit 26 employed in the optical magnetic disc apparatus 3 shown in Fig. 11 except that the subtractor 174 employed by the former is used as a substitute for the adder 74 of the latter.

Fig. 16 is a diagram showing a detailed configuration of the 1394 interface unit 36 which is shown in Fig. 15 in a simple and plain manner. It is also obvious that the configuration of the 1394 interface unit 36 employed in the DVD player 1 shown in Fig. 16 is basically identical with that of the 1394 interface unit 26 employed in the optical magnetic disc apparatus 3 shown in Fig. 12 except that the subtractor 174 employed by the former is used as a substitute for the adder 74 of the latter. An additive generator 171, an LFSR 172, a shrink generator 173, an adder 181, a register 182, an adder 183, a register 184, a register 185, an adder 186, a condition judging unit 191 and a FIFO unit 192 employed in the 1394 interface unit 36 of the optical magnetic disc apparatus 3 shown in Fig. 16 correspond to the additive generator 71, the LFSR 72, the shrink generator 73, the adder 81, the register 82, the adder 83, the register 84, the register 85, the adder 86, the condition judging unit 91 and a FIFO unit 92 employed in the 394 interface unit 26 of the DVD player 1 shown in Fig. 12 respectively.

Thus, since the operation of the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown

in Fig. 16 is basically the same as that of the 394 interface unit 26 employed in the DVD player 1 shown in Fig. 12, its explanation is not repeated. It should be noted, however, that the former is different from the latter in that, in the case of the former, the subtractor 174 subtracts the m-bit decryption key received from the FIFO unit 192 employed in the shrink generator 173 from an encrypted text, that is, m-bit data received from the DVD player 1 by way of the 1394 serial bus 11, to decrypt the encrypted text into the clear text.

In the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3, encrypted data is decrypted only once by using a session key S, which comprises an initial value key Ss and a derangement key Si, and a time variable key i, strictly speaking, the key i', as described above.

In the case of the personal computer 2, on the other hand, encrypted data is decrypted by the 1394 interface unit 49 using a time variable key i which is generated by the derangement key Si of the session key S and a key i' at a 1st stage of decryption and then further decrypted by the application unit 61 using an initial value key Ss of the session key S at a 2nd stage of decryption.

Fig. 17 is a diagram showing a typical configuration of the 1394 interface unit 49 employed in the personal computer 2 for decrypting the encrypted data or the encrypted text received from the DVD player 1 by way of the 1394 serial bus 11 by means of hardware. As shown in the figure, much like the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig.15 and the 1394 interface unit 26 employed in the DVD player 1 shown in Fig.II, the configuration comprises an additive generator 271, an LFSR (Linear Feedback Shift Register) 272, a shrink generator 273 and a subtractor 274 which correspond to the additive generator 171, the LFSR (Linear Feedback Shift Register) 172, the shrink generator 173 and the subtractor 174 shown in Fig. 15 respectively. The key i' for generating the time variable key i and the derangement key Si of the session key S for deranging the time variable key i input to the 1394 unit 49 shown in Fig. 17 from the license manager 62 are the same as the key i' and the derangement key Si input to the 1394 interface unit 36 shown in Fig. 15 from the firmware 30. However, all bits of the initial value key Ss of the session key S input to the 1394 unit 49 shown in Fig. 17 are reset to 0.

Fig. 18 is a diagram showing a detailed configuration of the 1394 interface unit 49 which is shown in Fig. 17 in a simple and plain manner. It is also obvious that the configuration of the 1394 interface unit 49 employed in the personal computer 2 shown in Fig. 18 is basically identical with that of the 1394 interface unit 26 employed in the DVD player 1 shown in Fig. 12 and the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig. 16 except that, in the case of the 1394 interface unit 49 shown in Fig. 18, since all bits of the initial value key Ss of the session key S input to the 1394 unit 49 shown in Fig. 17 are reset to 0, in essence, the decryption key is generated only from the time variable key i which is generated from the key i' and the derangement key Si as if the initial value key Ss were not available. As a result, at the subtractor 274, the encrypted data or the encrypted text is decrypted by using only the time variable key i. Since the initial value key Ss has not been used in the decryption yet, a completely clear text has not been obtained yet as a result of the decryption. That is to say, the result of the decryption is still in an encrypted state. Thus, data resulting from the decryption can not be used as it is even if the data is copied from the internal bus 51 to a hard disc mounted on the hard disc drive 47 or another recording medium.

Then, the data or the text decrypted by hardware in the 1349 interface unit 49 by using the time variable key i is further decrypted by software in the application module 61. Fig. 19 is a diagram showing a typical configuration of the application module 61. Basically resembling the 1394 interface unit 26 employed in the DVD player 1 shown in Fig. 11, the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig.15 and the 1394 interface unit 49 employed in the personal computer 2 shown in Fig. 17, the application module 61 shown in Fig. 19 comprises an additive generator 371, an LFSR (Linear Feedback Shift Register) 372, a shrink generator 373 and a subtractor 374 which have configurations identical with the additive generator 171, the LFSR (Linear Feedback Shift Register) 172, the shrink generator 173 and the subtractor 174 shown in Fig. 15 respectively.

It should be noted, however, that while the initial value key Ss of the session key S is supplied to the application module as is the case with the 1394 interface unit 26 employed in the DVD player 1 shown in Fig. 11 and the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig.15, the derangement key Si of the session key S for deranging the time variable key i and the key i' are each a unit element will all bits thereof reset to 0.

Fig. 20 is a diagram showing a detailed configuration of the application module 61 which is shown in Fig. 19 in a simple and plain manner. It is also obvious that the configuration of the application module 61 is basically identical with that of the 1394 interface unit 26 employed in the DVD player 1 shown in Fig. 12, the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig. 16 and the 1394 interface unit 49 employed in the personal computer 1 shown in Fig. 18. Components employed in the application module 61 shown in detail in Fig. 20, from the adder 381 employed in the additive generator 371 to the FIFO unit 392 employed in the shrink generator 373, correspond to the components employed in the 1394 interface unit 36 shown in Fig. 16, from the adder 181 employed in the additive generator 171 to the FIFO unit 192 employed in the shrink generator 173 respectively. Since all the bits of the key i' held in a register 384 and the derangement key Si held in a register 385 are 0, however, the bits of the time variable key i generated by the adder 386 are all 0. As a result, the application module 61 in essence operates as if the time variable key i were not present. That is to say, the generation of a decryption key is based only on the

initial value key Ss. Then, a subtractor 374 decrypts the encrypted data or by using the decryption key generated in this way to produce a clear text. As described above, the encrypted data is a result of the decryption carried out by the 1394 interface unit 49 based on the time variable key i, which is generated from the key i' and the derangement key Si, at the so called 1st stage of decryption. On the other hand, the decryption carried out by the application module 61 based on the initial value key Ss is called a 2nd stage of decryption for producing a final completely clear text.

When the decryption of the encrypted text described above is completed at the optical magnetic disc 3, the CPU 31 supplies the decrypted data to the drive 35 for recording the data onto an optical magnetic disc.

In the personal computer 2, on the other hand, the CPU 41 supplies the decrypted data resulting from the 1st stage of decryption carried out by the 1394 interface unit 49 typically to the hard disc drive 47 for recording the data by way of the internal bus 51. It should be noted that, in the personal computer 2, a predetermined board can be connected to the input/output interface unit 44 as the expansion board 48 for monitoring data transmitted through the internal bus 51 as described earlier. Nevertheless, it is only the application module 61 that is capable of finally decrypting data transmitted through the internal bus 51. Thus, even if the expansion board 48 is capable of monitoring encrypted data resulting from the decryption carried out by the 1394 interface unit 49 based on the time variable key i, the encrypted data is not the completely clear text because the data has not been decrypted by the application module 61 by using the initial value key Ss of the session key S. As a result, it is possible to prevent a completely clear text from being copied illegally provided that the completely clear text resulting from the final decryption carried out by the application module 61 is never transmitted through the internal bus 51.

Typically, adoption of the Diffie - Hellman technique allows the session key S to be shared by a source and sinks.

It is worth noting that there are cases in which the 1394 interface unit 49 or the application module 61 employed in the personal computer 2 has a relatively low processing power so that it is not capable of carrying out decryption of data. In order to cope with such a problem, either of the initial value key Ss of the session key S and the time variable key i or both can be generated in the source as a unit element. By the same token, by using either or both of the keys as a unit element in the sink, data can virtually be transmitted from the source to the sink without using the initial value key Ss of the session key S and the time variable key i. With such a scheme, however, it is more quite within the bounds of possibility that the data is copied illegally.

If the application module 61 itself is an illegal copy, it is much to be feared that the clear text resulting from decryption carried out by the application module 61 will also be copied illegally. In order to solve this problem, the license manager 62 may authenticate the application module 61 prior to decryption as described earlier.

As a method for authenticating the application module 61, a disclosed encryption key encryption method can be adopted in addition to the common session key encryption / decryption technique described earlier.

The configurations shown in Figs. 11, 12 and 15 to 20 satisfy a homomorphism relation. Thai is to say, if keys K_1 and K_2 are elements of a Galois field G, a group processing result $K_1 \cdot K_2$ of the two elements is also an element of the Galois field G. In addition, with respect to a predetermined function H, the following equation holds true.

$$H(K_1 \cdot K_2) = H(K_1) \cdot H(K_2)$$

Fig. 21 is a diagram showing another typical detailed configuration of the 1394 interface unit 26 employed in the DVD player 1. As shown in the figure, the initial value key Ss of the session key S is supplied to LFSRs 501 to 503 to be set therein as initial values. The widths of the LFSRs 501 to 503 are n_1 to n_3 bits respectively which are of the order of 20 bits. The LFSRs 501 to 503 are designed so that their widths n_1 to n_3 form an element in conjunction with each other. That is to say, for example, the high order n_1 bits, the intermediate order n_2 bits and the low order n_3 bits of the initial value key Ss of the session key S are set in the LFSRs 501, 502 and 503 respectively each as an initial value.

When an enable signal with the logic value 1 is supplied to the LFSRs 501 to 503 from a clocking function unit 506, the LFSRs 501 to 503 each shift the contents thereof by m bits, outputting m-bit data. The value of m can be set typically at 8, 16, 32 or 40.

The data output by the LFSR 501 is added to the data output by the LFSR 502 by an adder 504. A carry of the result of the addition carried out by the adder 504 is supplied to the clocking function unit 506 and the result of the addition itself is added to the data output by the LFSR 503 by an adder 505. A carry of the result of the addition carried out by the adder 504 is also supplied to the clocking function unit 506 and the result of the addition itself is supplied to an exclusive logical sum computing circuit 508.

The combination of the carries supplied by the adders 504 and 505 to the clocking function unit 506 is either 00, 01, 10 or 11. The clocking function unit 506 outputs data representing one of combinations 000 to 111 to the LFSRs 501 to 503 in accordance with the combination of the carries received from the adders 504 and 505. As described above, when the enable signal with the logic value 1 is supplied to the LFSRs 501 to 503 from the clocking function unit 506, the LFSRs 501 to 503 each shift the contents thereof by m bits, outputting new m-bit data. When the enable signal with the logic value 0 is supplied to the LFSRs 501 to 503 from the clocking function unit 506, on the other hand,

the LFSRs 501 to 503 do not shift the contents thereof, outputting the same m-bit data as the data output right before.

The exclusive logical sum computing circuit 508 receives the result of addition carried out by the adder 505 and the time variable key i stored in the register 507, calculating an exclusive logical sum of the inputs. An exclusive logical sum computing circuit 509 calculates another exclusive logical sum of the exclusive logical sum output by the exclusive logical sum computing circuit 508 and an input clear text, outputting the other exclusive logical sum as an encrypted text.

Fig. 22 is a diagram showing another typical detailed configuration of the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3. As shown in the figure, all components employed in the 1394 interface unit 36, from an LFSR 601 to an exclusive logical sum computing circuit 609, have the same configurations as the corresponding components employed in the 1394 interface unit 26 shown in Fig. 21, from the LFSR 501 to the exclusive logical sum computing circuit 509. Thus, since their operations are basically also the same, the explanation of their operations is not repeated. The only difference between the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig. 22 and the 1394 interface unit 26 employed in the DVD player 1 shown in Fig. 21 is that the exclusive logical sum computing circuit 609 employed in the former decrypts an encrypted text while the exclusive logical sum computing circuit 509 employed in the latter encrypts a clear text.

Fig. 23 is a diagram showing another typical detailed configuration of the 1394 interface unit 49 employed in the personal computer 2. As shown in the figure, all components employed in the 1394 interface unit 49, from an LFSR 701 to an exclusive logical sum computing circuit 709, have the same configurations as the corresponding components employed in the 1394 interface unit 36 shown in Fig. 22, from the LFSR 601 to the exclusive logical sum computing circuit 609. The only difference between the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig. 22 and the 1394 interface unit 49 employed in the personal computer 2 shown in Fig. 23 is that the initial value key Ss of the session key S supplied to the LFSRs 701 to 703 employed in the latter is a unit element will all bits thereof reset to 0. Thus, in the case of the 1394 interface unit 49 employed in the personal computer 2 shown in Fig. 23, the decryption of an encrypted text is in essence based only on the time variable key i in the register 707 which is generated from the key i' and the derangement key Si of the session key S.

Fig. 24 is a diagram showing another typical detailed configuration of the application module 61 of the personal computer 2. As shown in the figure, all components employed in the application module 61, from an LFSR 801 to an exclusive logical sum computing circuit 809, have the same configurations as the corresponding components employed in the 1394 interface unit 36 shown in Fig. 22, from the LFSR 601 to the exclusive logical sum computing circuit 609. The only difference between the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig. 22 and the application module 61 of the personal computer 2 shown in Fig. 24 is that the time variable key i supplied to the register 807 employed in the latter is a unit element will all bits thereof reset to 0. Thus, in the case of the application module 61 employed in the personal computer 2 shown in Fig. 24, the decryption of encrypted data is in essence based only on the initial value key Ss of the session key S.

It should be noted that the decryption processing in each of the configurations shown in Figs. 19, 20 and 24 is carried out by the application module 61 which is typically implemented by software.

By the way, a license key can be changed or updated, if necessary, should the license key be stolen for some reasons by any chance. It is needless to say that a license key can also be changed once a predetermined period of time even if the license key is not stolen should it be quite within the bounds of possibility that the license key is stolen. In this case, the version of a license key representing the term of validity is recorded on a DVD. In the case of the present embodiment, the term of validity of a license key is represented by the number of times the hash function is to be applied to generate the license key. If an information receiving apparatus for receiving information transmitted through a satellite instead of information played back from a DVD player is an object being operated, only information of a valid version is transmitted to the information receiving apparatus by way of the satellite.

Figs. 25 and 26 are diagrams showing an embodiment implementing a procedure for generating a source side session key sk in the DVD player 1 and a sink side session key sk' in the personal computer 2 by using an updated license key. It should be noted that, in addition to the fact that various pieces of information are stored in the EEPROM unit 27 employed in the DVD player 1 and the EEPROM unit 50 employed in the personal computer 2 of the embodiment shown in Fig. 4, the hash function is also stored not only in the EEPROM unit 26, but also in the EEPROM unit 50 in the case of the present embodiment.

As shown in Fig. 25, the procedure begins with a step S151 at which the DVD player 1 serving as a source makes a request to the personal computer 2 serving as a sink for the ID thereof. Then, the procedure goes on to a step S152 at which the personal computer 2 receives the request for the ID. The procedure then proceeds to a step S153 at which the personal computer 2 transmits the ID to the DVD player 1. Then, the procedure continues to a step S154 at which the DVD player 1 receives the ID.

Subsequently, the procedure goes on to a step S155 at which the DVD player 1 concatenates the ID received from the personal computer 2 with a service key stored in the EEPROM unit 27 to form data (ID II service_key). Then, a license key lk is computed by applying the hash function to the data (ID II service_key) as shown in the following equation:

$$lk = hash (ID \| service\_key)$$

The pieces of processing performed at the steps S151 to S155 as described above are the same as those carried out at the steps S1 to S5 of the procedure shown in Fig.4.

The procedure then goes on to a step S156 at which the DVD player 1 forms a judgment as to whether or not the license key lk generated at the step S155 has a valid version, that is, whether or not the license key lk has been generated by applying the hash function a number of times equal to a predetermined value recorded on the DVD. As described above, the present valid version of a license key lk is recorded as the predetermined value representing the number of times the hash function is to be applied to generate the license key lk. Assume that the predetermined value recorded on the DVD is greater than one. Since the number of times the hash function has been applied to generate the license key lk at the step S155 is 1, the license key lk is judged to be invalid. In this case the procedure proceeds to a step S157 at which the DVD player 1 initializes a variable g indicating the number of times the hash function has been applied to generate the license key lk at 1 and stores the generated license key lk in a variable lk_g. Then, the procedure continues to a step S158 at which the hash function is applied to the contents of the variable lk_g to find a new license key lk_{g+1} according to the following equation:

$$lk_{g+1} = hash (lk_g)$$

Subsequently, the procedure goes on to a step S159 to form a judgment as to whether or not the license key lk_{g+1} generated at the step S158 has a valid version. If the license key lk_{g+1} does not have a valid version, that is, if the variable g has not reached the predetermined value, the procedure proceeds to a step S160 at which the DVD player 1 increments the value of the variable g by 1 and stores lk_{g+1} in the variable lk_g. The procedure then returns to the step S158 at which the hash function is again applied to the contents of the variable lk_g.

The steps S158 and S159 are executed repeatedly till the value of the variable g, that is, the number of times the hash function has been applied to generate the license key, reaches the predetermined value recorded on the DVD as a version of the license key.

It should be noted that the predetermined value serving as an upper limit of the number of times the hash function can be applied to generate the license key is set typically at 100.

If the outcome of the judgment formed at the step S159 indicates that the number of times the hash function has been applied to generate the license key has reached the predetermined value recorded on the DVD as a version of the license key, that is, if the outcome of the judgment indicates that a valid license key lk_{g+1} has been obtained at the step S158, or if the outcome of the judgment formed at the step S156 indicates that the license key lk generated at the step S155 is valid, that is, if the number of times the hash function is to be applied to generate the license key is 1, on the other hand, the procedure proceeds to a step S161 at which the DVD player 1 generates a source side session key sk in the same way as the procedure of Fig. 4 described earlier.

Then, the procedure continues to a step S162 at which the DVD player 1 encrypts the source side session key sk generated at the step S161 by using the license key lk_g computed at the step S155 or S158 as a key to produce an encrypted source side session key e in accordance with the following equation:

$$e = Enc (lk_g, sk)$$

Subsequently, the procedure goes on to a step S163 at which the DVD player 1 transmits the encrypted source side session key e generated at the step S162 along with the value of the variable g indicating the number of times the hash function has been applied to generate the license key lkg to the personal computer 2. The procedure then proceeds to a step S164 at which the personal computer 2 receives the encrypted source side session key e and the value of the variable g. Then, the procedure proceeds to a step S165 at which the personal computer 2 initializes a variable w representing the number of times the hash function has been applied to generate a license key in the personal computer 2 at 1. The procedure then continues to a step S166 to form a judgment as to whether or not the value of the variable g received at the step S164 is equal to the value of the variable w set at the step S165. If they are not equal to each other, the procedure goes on to a step S167 at which the hash function stored in the EEPROM unit 50 employed in the personal computer 2 is applied to license_keyw, the license key also stored in the EEPROM unit 50, to generate $license\_key_{w+1}$, a new license key in accordance with the following equation:

$$license\_key_{w+1} = hash (license\_key_w)$$

Then, the procedure continues to a step S168 at which the personal computer 2 increments the variable w by 1 and substitutes $license\_key_{w+1}$ for $license\_key_w$. The procedure then returns to the step S166 to again form a judgment as to whether or not the value of the variable g is equal to the value of the variable w. The steps S166 to S168 are executed repeatedly till the value of the variable w representing the number of times the hash function has been applied to generate the license key becomes equal to the value of the variable g.

If the outcome of the judgment formed at the step S166 indicates the value of the variable w is equal to the value of the variable g, that is, if currently valid $license\_key_w$ has been obtained, the procedure goes on to a step S169 at which the personal computer 2 decrypts the encrypted source side session key e to produce a sink side session key sk' in accordance with the following equation:

$$sk' = Dec\ (license\_key_w,\ e)$$

By appropriately repeating the application of the hash function to generate the license key as described above, the information security can be further enhanced.

According to the procedure shown in Figs. 25 and 26, the value of the variable g representing the version of a license key is transmitted by the source to the sink. It should be noted, however, that the application of the hash function to generate the license key can be repeated as many times as is required without the need to transmit the version as is the case with an embodiment implementing a procedure shown in Fig. 25 and continued to Fig. 27 instead of Fig. 26.

That is to say, in the case of this embodiment, only the encrypted source side session key e is transmitted by the DVD player 1 to the personal computer 2 at the step S163. At that time, the value of the variable g representing the version of a license key is not transmitted. The procedure then proceeds to a step S164 at which the personal computer 2 receives the encrypted source side session key e. Then, the procedure goes on to a step S165 at which the personal computer 2 decrypts the encrypted source side session key e to produce a sink side session key sk' using the license key stored in the EEPROM unit 50 in accordance with the following equation:

$$sk' = Dec\ (license\ key,\ e)$$

In the mean time, at a step S166, the DVD player 1 encrypts data to be transmitted to the personal computer 2 by using, among other keys, the source side session key sk generated at the step S161 and transmits the encrypted data to the computer 2. The procedure then goes on to a step S167 at which the personal computer 2 receives the encrypted data and then to a step S168 to decrypt the encrypted data by using, among other keys, the sink side session key sk' generated at the step S165. Then, the procedure proceeds to a step S169 at which the personal computer 2 forms a judgment as to whether or not data resulting from the decryption carried out at the step S168 is correct. For example, data received as a TS (Transport Stream) packet of the MPEG system has a code for synchronization with a hexadecimal value of 47 in the head of the packet. In this case, the judgment as to whether or not data is correct can be formed by checking whether or not the synchronization code is perfect.

If correct decrypted data was not resulted in at the step S168, the procedure goes on to a step S170 at which the personal computer 2 updates the license key in accordance with the following equation:

$$license\_key = hash\ (license\_key)$$

Then, the procedure proceeds to a step S171 at which the personal computer 2 again decrypts the encrypted source side session key e received at the step S164 to produce a new sink side session key sk' using the updated license key generated at the step S170 in accordance with the following equation:

$$sk' = Dec\ (license\_key,\ e)$$

Subsequently, the procedure returns to the step S168 to again decrypt the encrypted data received at the step S167 by using, among other keys, the sink side session key sk' generated at the step S171. Then, the procedure proceeds to a step S169 at which the personal computer 2 forms a judgment as to whether or not data resulting from the decryption carried out at the step S168 is correct. As such, the steps S170, S171, S168 and S169 are executed repeatedly till the outcome of the judgment formed at the step S169 indicates that correct decrypted data was obtained at the step S168.

In this way, the license key is updated to produce correct encrypted data.

As indicated by the procedure described above, in the source, the source side session key sk has to be generated before data to be transmitted to the sink is encrypted by using the source side session key sk. In the sink, on the other hand, the decryption of the encrypted data received from the source needs to be synchronized with the decryption of the encrypted source side session key e received from the source. To be more specific, the procedure on the sink side can not go on from the step S165 to decrypt the encrypted source side session key e to the step S168 to decrypt the decrypted data till the step S167 to receive the encrypted data is completed.

In addition, the decryption of an encrypted source side session key e and an encrypted text carried out by the sink must be synchronized with the encryption of a source side session key sk and a clear text performed by the source. That is to say, a decryption key generated by the components composing the 1394 interface unit 36 employed in the optical magnetic disc apparatus 3 shown in Fig. 22, from the LFSR 601 to the exclusive logical sum computing circuit 608, has to correspond to an encryption key generated by the components composing the 1394 interface unit 26 employed in the DVD player 1 shown in Fig. 21, from the LFSR 501 to the exclusive logical sum computing circuit 508, and encrypted data decrypted by using the decryption key must be data resulting from encryption of a clear text by using the encryption key. As described above, the encryption key has thus to be generated by the 1394 interface unit 26 shown in Fig. 21 in synchronization with (that is, prior to) the encryption of the input clear text and the decryption key must therefore be generated by the 1394 interface unit 36 shown in Fig. 22 in synchronization with (that is, prior to) the decryption of the received encrypted text even though the synchronization is not explicitly shown in Figs. 21 and 22.

Accordingly, if a bit is missing for some reasons from a packet composing an encrypted text transmitted from a source to a sink by way of the 1394 serial bus 11, a phase representing a timing relation between a clear text and an encryption key in the source can not be sustained as a phase representing a timing relation between an encrypted text and a decryption key in the sink. However, this problem can be solved by updating or reinitializing the phase representing a timing relation between an encrypted text and a decryption key in the sink periodically. Fig. 28 is a diagram showing a typical configuration of an embodiment implementing a source / sink system for updating or reinitializing the phase representing a timing relation between an encrypted text and a decryption key in the sink periodically.

As shown in the figure, in the source, an exclusive logical sum computing circuit 901 computes an exclusive logical sum Ci of a random number generated by a random number generator 903 and an input clear text and outputs the exclusive logical sum Ci to an exclusive logical sum computing circuit 904 and a processing circuit 902 which also receives the initial value key Ss of a session key S. The processing circuit 902 carries out predetermined processing on the initial value key Ss of the session key S and the exclusive logical sum Ci output by the exclusive logical sum computing circuit 901, outputting a result Vi of the processing to the random number generator 903 as an initial value.

The exclusive logical sum computing circuit 904 computes the exclusive logical sum of the exclusive logical sum Ci generated by the exclusive logical sum computing circuit 901 and a time variable key i to generate an encrypted text which is transmitted to the sink through the 1394 serial bus 11.

The sink carries out operations in the reversed order of those performed by the source. To be more specific, an exclusive logical sum computing circuit 911 computes an exclusive logical sum Ci of the encrypted text received from the source through the 1394 serial bus 11 and the time variable key i, outputting the exclusive logical sum Ci to an exclusive logical sum computing circuit 912 and a processing circuit 913 which also receives the initial value key Ss of the session key S. The processing circuit 913 carries out predetermined processing on the initial value key Ss of the session key S and the exclusive logical sum Ci output by the exclusive logical sum computing circuit 911, outputting a processing result Vi to a random number generator 914. The random number generator 914 generates a random number with the processing result Vi from the processing circuit 913 used as an initial value. The exclusive logical sum computing circuit 912 computes a final exclusive logical sum of the random number generated by the random number generator 914 and the exclusive logical sum Ci generated by the exclusive logical sum computing circuit 911, outputting the final exclusive logical sum as a clear text.

Fig. 29 is a diagram showing a typical configuration of the random number generator 903. As shown in the figure, the random number generators 903 comprises components, from an LFSR 931 to a clocking function unit 936. Each of the components shown in the figure has a function identical with the corresponding LFSR 501 etc., the adder 504 etc. or the clock functioning unit 506 etc. of the embodiments shown in Figs. 21 to 24.

It should be noted that the random number generator 914 has the same configuration as the random number generator 903 shown in Fig. 29. Therefore, it is not necessary to show the configuration of former in a separate figure.

Fig. 30 shows a flowchart representing operations carried out by each of the processing circuits 902 and 913 on the source and sink sides respectively.

The operations are explained by referring to the flowchart shown in Fig. 30 as follows.

The processing circuit 902 on the source side has a function f expressed by an equation given below to compute a value Vi from an input Ci supplied thereto by the exclusive logical sum computing circuit 901 and the initial value key Ss of a session key S.

$$Vi = f(Ss, Ci)$$

As shown in the figure, the flowchart begins with a step S201 at which the processing circuit 902 uses the value 0 as an initial value of the input Ci to compute a value Vi = f (Ss, Ci) as follows:

$$V_0 = f(Ss, 0)$$

The operational flow then goes on to a step S202 at which the value V0 computed at the step S201 is supplied to the random number generator 903 shown in Fig. 29. In the random number generator 903, the value V0 output by the processing circuit 902 is supplied to the LFSR 931 to 933 as an initial value. By using the same technique as the 1394 interface unit 26 shown in Fig. 21 and the other embodiments shown in Figs. 22 to 24, a random number is generated and output by the adder 935 employed in the random number generator 903 to the exclusive logical sum computing circuit 901 shown in Fig. 28. The exclusive logical sum computing circuit 901 computes an exclusive logical sum Ci of the random number generated by the random number generator 903 and an input clear text, outputting the exclusive logical sum Ci back to the processing circuit 902.

In the mean time, the operational flow shown in Fig. 30 proceeds to a step S203 at which the processing circuit 902 sets a variable i at 1. The operational flow then continues to a step S204 at which the exclusive logical sum Ci received from the exclusive logical sum computing circuit 901 is stored in a variable C.

Then, the operational flow goes on to a step S205 at which the processing circuit 902 carries out processing in accordance with the following equation:

$$Vi = f(Ss, Ci) + V_{i-1}$$

where Ci is the contents of the variable C.

Since the value of the variable i is 1 at the present time, the above equation can be rewritten as follows:

$$V1 = f(Ss, C_1) + V_0$$

where V0 is a value computed at the step S201.

Subsequently, the operational procedure goes on to a step S206 at which the processing circuit 902 forms a judgment as to whether or not the contents of the variable C, that is, C_1 in this case, are equal to a predetermined value T set in advance. In the mean time, the exclusive logical sum computing circuit 901 outputs other exclusive logical sum Ci to the processing circuit 902. If the exclusive logical sum Ci is found unequal to the value T at the step S206, the operational flow proceeds to a step S207 at which the contents of the variable i are incremented by 1 before returning to the step S204 at which the other exclusive logical sum Ci received from the exclusive logical sum computing circuit 901, that is, C_2 since i = 2, is stored in the variable C.

Then, the operational flow goes on to the step S205 at which the processing circuit 902 carries out processing in accordance with the following equation:

$$V_2 = f(Ss, C_2) + V_1$$

where V1 is a value computed at the step S205 in the immediately previous iteration.

Subsequently, the operational procedure goes on to the step S206 at which the processing circuit 902 forms a judgment as to whether or not the input exclusive logical sum Ci, that is, C2 in this case, is equal to the predetermined value T. If the input exclusive logical sum Ci is found unequal to the value T, the operational flow proceeds to the step S207 at which the contents of the variable i are incremented by 1 before returning to the step S204. In this way, the steps S204 to S207 are executed repeatedly till the input exclusive logical sum Ci becomes equal to the value T.

If the input exclusive logical sum Ci is found equal to the value T at the step S206, on the other hand, the operational flow proceeds to the step S208 at which the value Vi (that is, V1 in this case) computed at the step S205 is output to the random number generator 903 as the value V0 computed at the step S201 was output to the random number generator 903 at the step S202. In the random number generator 903, the value V1 output by the processing circuit 902 is supplied to the LFSR 931 to 933 as an initial value. A random number is generated and output by the adder 935 employed in the random number generator 903 to the exclusive logical sum computing circuit 901 shown in Fig. 28.

The exclusive logical sum computing circuit 901 computes an exclusive logical sum Ci of the random number generated by the random number generator 903 and an input clear text, outputting the exclusive logical sum Ci back to the processing circuit 902.

In the mean time, after the processing circuit 902 outputs the value Vi at the step S208 to the random number generator 903, the operational flow shown in Fig. 30 returns to the step S203 at which the processing circuit 902 resets the variable i at 1. Thereafter, the steps S203 to S208 are executed repeatedly.

Assume that the value T is 8 bits in width and the generation probability of the value of Ci is uniform. In this case, the probability of the Ci value's being equal to T is 1/256 where 256 is the eighth power of 2. That is to say, the generation of the exclusive logical sum Ci having a value equal to T occurs at a rate of once per 256 sequential operations carried out by the exclusive logical sum computing circuit 901 to generate the exclusive logical sum Ci. As a result, the initial value used in the random number generator 903 for generating a random number is updated at a rate of once per 256 sequential operations carried out by the exclusive logical sum computing circuit 901 to generate the exclusive logical sum Ci.

The exclusive logical sum Ci output by the exclusive logical sum computing circuit 901 is also supplied to the exclusive logical sum computing circuit 904 for computing the exclusive logical sum of the exclusive logical sum Ci and the time variable key i. The exclusive logical sum computed by the exclusive logical sum computing circuit 904 is output to the 1394 serial bus 11 as an encrypted text.

In the sink, the exclusive logical sum computing circuit 911 computes an exclusive logical sum Ci of the encrypted text received from the source through the 1394 serial bus 11 and the time variable key i, outputting the exclusive logical sum Ci to the exclusive logical sum computing circuit 912 and the processing circuit 913 which also receives the initial value key Ss of the session key S. Much like the processing circuit 902 on the source side, the processing circuit 913 carries out predetermined processing on the initial value key Ss of the session key S and the exclusive logical sum Ci output by the exclusive logical sum computing circuit 911, outputting a processing result Vi to the random number generator 914 at a rate of once per 256 sequential operations to generate the exclusive logical sum Ci. The random number generator 914 generates a random number with the processing result Vi used as an initial value. The exclusive logical sum computing circuit 912 computes a final exclusive logical sum of the random number generated by the random number generator 914 and the exclusive logical sum Ci generated by the exclusive logical sum computing circuit 911 and outputs the final exclusive logical sum as a clear text.

As described above, the processing circuit 913 outputs the processing result Vi to the random number generator 914 at a rate of once per 256 sequential operations carried out by the exclusive logical sum computing circuit 911 to generate the exclusive logical sum Ci. As a result, a phase representing a timing relation between an encrypted text transmitted from a source to a sink by way of the 1394 serial bus 11 and a random number used as a decryption key in the sink can be recovered in the event of a bit missing for some reasons from a packet composing the encrypted text at the time the processing circuit 913 outputs the processing result Vi to the random number generator 914 at a rate of once per 256 sequential operations to generate the exclusive logical sum Ci.

It should be noted that, since the processing circuit 902 or 913 outputs the processing result Vi to the random number generator 914 when the exclusive logical sum Ci becomes equal to the value T (Ci = T), the processing circuit 913 does not output the processing result Vi to the random number generator 914 periodically. Instead, nothing more can be said more than the fact that the processing circuit 913 outputs the processing result Vi to the random number generator 914 at a probability of once per 256 sequential operations to generate the exclusive logical sum Ci on the average.

It is worth noting that the rate at which the processing circuits 902 and 913 output the processing result Vi to the random number generators 903 and 914 can also be based on the number of pieces of encrypted data transmitted by the source and received by the sink. When a piece of data is missing in the course of transmission through the 1394 serial bus 11, however, this method will have a problem that the data piece count on the source side will be different from the data piece count on the sink side, making it no longer possible to establish synchronization between the source and the sink. It is thus desirable to adopt the synchronization technique implemented by the embodiment described above.

As an initial value used in the random number generator 903 or 914, the exclusive logical sum Ci output by the exclusive logical sum computing circuit 901 or 911 can be supplied to the random number generator 903 or 914 respectively as it is. In this case, however, transmitted through the 1394 serial bus 11, it is much to be feared that the exclusive logical sum Ci is stolen. That is why the exclusive logical sum Ci is not used directly as an initial value. Instead, by using a value Vi resulting from predetermined processing carried out on the exclusive logical sum Ci as an initial value, the data security can be further improved.

In the embodiment implementing an authentication procedure shown in Fig. 4, the license key sk is fixed. It should be noted, however, that the license key lk can be changed each time the authentication procedure is executed. Fig. 31 is a diagram showing an embodiment implementing an authentication procedure wherein the license key lk is changed each time the authentication procedure is executed.

As shown in Fig. 31, the procedure begins with a step S211 at which the firmware 20 in the DVD player 1 controls the 1394 interface unit 26 to make a request to the personal computer 2 for the ID thereof to be transmitted by way of the 1394 serial bus 11. Then, the procedure goes on to a step S212 at which the license manager 62 of the personal computer 2 receives the request for the ID. To put it in detail, the 1394 interface unit 49 employed in the personal computer 2 passes on the request for the ID transmitted by the DVD player 1 by way of the 1394 serial bus 11 to the CPU 41. The procedure then proceeds to a step S213 at which the license manager 62 being executed by the CPU 41 reads out the ID from the EEPROM unit 50 in accordance with the request forwarded thereto by the 1394 interface unit 49 and transmits it to the DVD player 1 by way of the 1394 interface unit 49 and the 1394 serial bus 11.

Then, the procedure continues to a step S214 at which the 1394 interface unit 26 employed in the DVD player 1 receives the ID and passes it to the firmware 20 being executed by the CPU 21.

Subsequently, the procedure goes on to a step S215 at which the firmware 20 concatenates the ID received from the personal computer 2 with a service key stored in the EEPROM unit 27 to form data (ID || service_key). Then, a license key lk is computed by applying the hash function to the data (ID || service_key) as shown in the following equation:

$$lk = \text{hash}(ID \,||\, service\_key)$$

The procedure then proceeds to a step S216 at which the firmware 20 generates a random number r. Then, the procedure proceeds to a step S217 at which the firmware 20 concatenates the license key lk with the random number r and modifies the license key lk to a license key lk' by applying the hash function to the result of concatenation as follows:

$$lk' = \text{hash}(lk \,||\, r)$$

Subsequently, the procedure proceeds to a step S218 at which the firmware 20 generates a source side session key sk. Then, the procedure continues to a step S219 at which the firmware 20 encrypts the source side session key sk generated at the step S218 by using the license key lk' computed at the step S217 as a key to an encrypted source side session key e in accordance with the following equation:

$$e = \text{Enc}(lk', sk)$$

Subsequently, the procedure goes on to a step S220 at which the firmware 20 transmits the encrypted source side session key e generated at the step S219 and the random number r generated at the step S216 to the personal computer 2. To put it in detail, the encrypted source side session key e and the random number r are transmitted by the 1394 interface unit 26 employed in the DVD player 1 to the personal computer 2 by way of the 1394 serial bus 11. The procedure then proceeds to a step S221 at which the 1394 interface unit 49 employed in the personal computer 2 receives the encrypted source side session key e and the random number r. Subsequently, the procedure goes on to a step S222 at which the license manager 62 generates a license key lk" by applying the hash function stored in the EEPROM unit 50 to a result of concatenation of the random number received at the step S221 with a license key stored in the EEPROM unit 50 as follows:

$$lk'' = \text{hash}(license\_key \,||\, r)$$

Then, the procedure proceeds to a step S223 at which the license manager 62 decrypts the encrypted source side session key e passed on thereto by the 1394 interface unit 49 by using the license key lk" generated at the step S222 as a key to produce a sink side session key sk' in accordance with the following equation:

$$sk' = \text{Dec}(lk'', e)$$

Since the license key given to the personal computer 2 by the author of information and stored in the EEPROM unit 50 was generated in the same way as the license key lk generated in the DVD player 1 at the step S215, the license key lk" generated by the personal computer 2 at the step S222 has the same value as the license key lk' generated in the DVD player 1 at the step S217. That is to say, the following equation holds true:

$$lk' = lk''$$

As a result, the sink side session key sk' resulting from the decryption of the encrypted source side session key e carried out by the personal computer 2 at the step S223 has the same value as the source side session key sk generated by the DVD player 1 at the step S218. That is to say, the following equation holds true:

$$sk' = sk$$

By changing the license key lk' used for encrypting the source side session key sk from time to time before transmitting the key sk to the sink, it is less to be feared that the encrypted source side session key sk transmitted to the sink can be decrypted by an unauthorized person who knows a fixed license key by any chance.

In the embodiments described above, the DVD player 1 serves as a source while the personal computer 2 and the optical magnetic disc apparatus 3 each serve as a sink. It should be noted that the description is not intended to be construed in a limiting sense. That is to say, any arbitrary electronic apparatus can be used as a source or a sink.

In addition, while the 1394 serial bus 11 is used as an external bus for connecting the electronic apparatuses composing a data processing system to each other, the scope of the present embodiment is not limited to such embodiments. That is, a variety of buses can be used as an external bus and electronic apparatuses connected to each other by the external bus are not limited to those employed in the embodiments described above. Any arbitrary electronic apparatuses can be used to compose the data processing system.

It is also worth noting that a variety of programs consisting of instructions to be executed by CPUs are presented to the user through providing media such a magnetic disc, a CD-ROM disc and a network and can be used, if necessary, by storing the programs in a RAM unit or a hard disc incorporated in the electronic apparatus.

According to an embodiment of a data transmitting apparatus as claimed in claim 1, an embodiment of a data transmitting method as claimed in claim 10 and an embodiment of a recording medium as claimed in claim 41, computation of a first value lk (or the so-called license key) is based on an ID received from other equipment and the apparatus' or the method's own ID (or the so-called service key used for identifying information to be processed or identifying a service for processing the information) as shown on the source side of the procedure of Fig. 4. As a result, the security of transmitted data can be improved for a reason described as follows.

To put it in detail, the first value lk is computed by the data transmitting apparatus or the data transmitting method by applying a predetermined method or a predetermined sub-method respectively to the ID received from other equipment and the apparatus' or the method's own ID. Key information sk is then generated and predetermined processing based upon the 1st value lk is further carried out on the key information sk. Finally, a result e of the predetermined processing is transmitted to the other equipment. As a result, only valid other equipment is allowed to carry out predetermined data processing, giving rise to an even improved security of the transmitted data.

In addition, according to an embodiment of a data transmitting apparatus as claimed in claim 1, an embodiment of a data transmitting method as claimed in claim 10 and an embodiment of a recording medium as claimed in claim 41 as well as an embodiment of a data receiving apparatus as claimed in claim 19, an embodiment of a data receiving method as claimed in claim 30 and an embodiment of a recording medium as claimed in claim 43, the 1st value lk is computed by the data transmitting apparatus or the data transmitting method by applying a predetermined method or a predetermined sub-method respectively to the data receiving apparatus' or the data receiving method's own ID transmitted by the data receiving apparatus or the data receiving method and the data transmitting apparatus' or the data transmitting method's own ID. Key information sk is then generated by the data transmitting apparatus or the data transmitting method and predetermined processing based upon the 1st value lk is further carried out on the key information sk by the data transmitting apparatus or the data transmitting method. Finally, a result e of the predetermined processing is transmitted to the data receiving apparatus wherein the result e of the predetermined processing is decrypted by the data receiving apparatus or the data receiving method by using a license key having the same value as the 1st value lk. As a result, an information processing system offering an even higher security of transmitted data can be implemented.

In another embodiment implementing the information processing system described above as shown in Fig. 9, a first value H is computed by the data transmitting apparatus or the data transmitting method by applying a predetermined hash function to the data receiving apparatus' or the data receiving method's own ID transmitted by the data receiving apparatus or the data receiving method and the data transmitting apparatus' or the data transmitting method's own ID. Key information sk is then generated by the data transmitting apparatus or the data transmitting method and predetermined processing based upon the 1st value H is further carried out on the key information sk by the data transmitting apparatus or the data transmitting method. Finally, a result e of the predetermined processing is transmitted to the data receiving apparatus wherein the result e of the predetermined processing is decrypted by using two license keys LK

and LK' provided to the data receiving apparatus or the data receiving method. The license keys LK and LK' are generated in advance typically by the author of information by using the predetermined hash function, a pseudo random number generating function pRNG, and the inverse function G^-1 of a confusion function G. By also applying the pseudo random number generating function pRNG in the predetermined processing and the pseudo random number generating function pRNG and the confusion function G in the decryption of the result e of the predetermined processing, the security of the transmitted data can be further improved for reasons described as follows.

To put it in detail, in the other embodiment described above, the result e of the predetermined processing is obtained by encryption of the key information sk using a pseudo random number pRNG(H) obtained from the 1st value H. As a result, the security of the transmitted data can be further improved by the more complicated processing.

In addition, the aforementioned license key LK' provided to the data receiving apparatus or the data receiving method is computed in advance by applying the inverse function G^-1 to a result R which is obtained by applying the pseudo random number generating function pRNG to the 1st value H and the license key LK. As a result, an information processing system offering an even better security of transmitted data can be implemented through the use of the license key LK' derived from a more complex calculation in addition to the license key LK.

Claims

1. A data transmitting apparatus wherein data is transmitted after predetermined processing based upon said apparatus' own ID code and an ID code received from other equipment has been carried out, said apparatus comprising:

    a storage means for storing said apparatus' own ID code;
    a reception means for receiving an ID code from other equipment;
    a 1st calculation means for calculating a 1st value by application of a predetermined method to an output of said storage means and an output of said reception means;
    a key information generating means for generating key information;
    an information processing means for carrying out said predetermined processing based upon said 1st value on said key information; and
    a transmission means for transmitting an output of said information processing means.

2. A data transmitting apparatus according to claim 1 wherein, in accordance with said predetermined method, a predetermined function is applied a predetermined number of times.

3. A data transmitting apparatus according to claim 2 wherein said predetermined function is modified in accordance with said predetermined number of times.

4. A data transmitting apparatus according to claim 3 wherein said predetermined function is a hash function.

5. An encryption apparatus for carrying out encryption based on a predetermined key code, said apparatus comprising:

    a random number generator for generating a random number from an input value;
    a synthesis means for merging said random number output by said random number generator with input data to generate encrypted data; and
    a processing means for carrying out processing based upon said predetermined key code on said encrypted data output by said synthesis means and supplying an output signal to said random number generator as said input value.

6. An encryption apparatus according to claim 5 wherein said processing means compares said encrypted data output by said synthesis means with a predetermined value and changes a parameter used in said processing till said encrypted data output at said synthesis means becomes equal to said predetermined value.

7. An encryption apparatus according to claim 5 further comprising a 2nd synthesis means for generating final encrypted data by encryption of said encrypted data output by said synthesis means based on a 2nd key code.

8. A data transmitting apparatus according to claim 1 wherein, in accordance with said predetermined method, a predetermined function is applied and a random number is used in application of said predetermined function.

9. A data transmitting apparatus according to claim 8 wherein said predetermined function is a hash function.

10. A data transmitting method whereby data is transmitted after predetermined processing based upon an own ID code and an ID code received from other equipment has been carried out, said method comprising the steps of:

    reading out said own ID code from a storage means;
    receiving an ID code transmitted by other equipment;
    calculating a 1st value by application of a predetermined sub-method to said ID read out from said storage means and said ID received from said other equipment;
    generating key information;
    carrying out said predetermined processing based upon said 1st value on said key information; and
    transmitting said key information completing said predetermined processing.

11. A data transmitting method according to claim 10 wherein, in accordance with said predetermined sub-method, a predetermined function is applied a predetermined number of times.

12. A data transmitting method according to claim 11 wherein said predetermined function is modified in accordance with said predetermined number of times.

13. A data transmitting method according to claim 12 wherein said predetermined function is a hash function.

14. An encryption method for carrying out encryption based on a predetermined key code, said method comprising:

    a random number generating step of generating a random number from an input value;
    a data merging step of merging said random number with input data to generate encrypted data; and
    a data processing step of carrying out processing based upon said predetermined key code on said encrypted data generated at said data merging step and supplying an output signal to said random number generating step as said input value.

15. An encryption method according to claim 14 whereby, at said data processing step, said encrypted data output at said data merging step is compared with a predetermined value and a parameter used in said processing is changed till said encrypted data generated at said data merging step becomes equal to said predetermined value.

16. An encryption method according to claim 14 whereby final encrypted data is generated by encryption of said encrypted data generated at said data merging step based on a 2nd key code.

17. A data transmitting method according to claim 10 whereby, in accordance with said predetermined sub-method, a predetermined function is applied and a random number is used in application of said predetermined function.

18. A data transmitting method according to claim 17 wherein said predetermined function is a hash function.

19. A data receiving apparatus for decrypting data received from other equipment by decryption based on said apparatus' own key and information received from said other equipment, said apparatus comprising:

    a storage means for storing said apparatus' own key;
    a reception means for receiving data and information from other equipment; and
    a decryption means for decrypting an output of said reception means by using an output of said storage means as a base.

20. A data receiving apparatus according to claim 19 wherein a judgment as to whether or not said apparatus' own key satisfies a predetermined condition is formed and predetermined processing is carried out on said apparatus' own key till said apparatus' own key satisfies said predetermined condition.

21. A data receiving apparatus according to claim 20 wherein:

    said predetermined condition is judged to be satisfied by said apparatus's own key if the number of times said predetermined processing is carried out on said apparatus' own key is equal to a predetermined value included in data received from other equipment; and

said predetermined processing is carried out on said apparatus' own key by applying a predetermined function to said apparatus' own key while incrementing the number of times said predetermined function is applied to said apparatus' own key.

22. A data receiving apparatus according to claim 20 wherein:

said predetermined condition is judged to be satisfied by said apparatus's own key if a result of decryption of data received from other equipment is correct; and
said predetermined processing is carried out on said apparatus' own key by applying a predetermined function to said apparatus' own key.

23. A data receiving apparatus according to claim 21 wherein said predetermined function is a hash function.

24. A data receiving apparatus according to claim 22 wherein said predetermined function is a hash function.

25. An encrypted data decrypting apparatus for carrying out decryption of encrypted data based on a predetermined key code, said apparatus comprising:

a random number generator for generating a random number from an input value;
a synthesis means for merging said random number generated by said random number generator with input data to generate a clear text; and
a processing means for carrying out processing based upon said predetermined key code on said input data and supplying an output signal as said input value to said random number generator.

26. An encrypted data decrypting apparatus according to claim 25 wherein said processing means compares said input value with a predetermined value and changes a parameter used in said processing till said input data becomes equal to said predetermined value.

27. An encrypted data decrypting apparatus according to claim 25 further comprising a 2nd synthesis means for generating said input data by decryption of said encrypted data based on a 2nd key code.

28. A data receiving apparatus according to claim 19 wherein said decryption is application of a predetermined function and a random number is used in said application of said predetermined function.

29. A data transmitting apparatus according to claim 28 wherein said predetermined function is a hash function.

30. A data receiving method for decrypting data received from other equipment by decryption based on an own key and information received from said other equipment, said method comprising the steps of:

reading out said method' own key from a storage means;
receiving data and information from other equipment; and
decrypting said data by using said key and said information as a base.

31. A data receiving method according to claim 30 wherein a judgment as to whether or not said method' own key satisfies a predetermined condition is formed and predetermined processing is carried out on said method' own key till said method' own key satisfies said predetermined condition.

32. A data receiving method according to claim 31 wherein:

said predetermined condition is judged to be satisfied by an own key if the number of times said predetermined processing is carried out on said own key is equal to a predetermined value included in data received from other equipment; and
said predetermined processing is carried out on said method' own key by applying a predetermined function to said method' own key while incrementing the number of times said predetermined function is applied to said method' own key.

33. A data receiving method according to claim 31 wherein:

said predetermined condition is judged to be satisfied by an own key if a result of decryption of data received from other equipment is correct; and
said predetermined processing is carried out on said method' own key by applying a predetermined function to said own key.

34. A data receiving method according to claim 32 wherein said predetermined function is a hash function.

35. A data receiving method according to claim 33 wherein said predetermined function is a hash function.

36. An encrypted data decrypting method for carrying out decryption of encrypted data based on a predetermined key code, said method comprising:

a random number generating step of generating a random number from an input value;
a random number merging step of merging said random number with input data to generate a clear text; and
a processing step of carrying out processing based upon said predetermined key code on said input data and supplying an output signal to said random number generating step as said input value for generating said random number.

37. An encrypted data decrypting method according to claim 36 wherein, in said processing, said input value is compared with a predetermined value and a parameter used in said processing is changed till said input data becomes equal to said predetermined value.

38. An encrypted data decrypting method according to claim 36 wherein said input data supplied to said random number merging step is generated by merging said encrypted data received from an external source with a 2nd key code.

39. A data receiving method according to claim 30 wherein said decryption is application of a predetermined function and a random number is used in said application of said predetermined function.

40. A data transmitting method according to claim 39 wherein said predetermined function is a hash function.

41. A recording medium for recording a program prescribing a data transmitting method whereby data is transmitted after predetermined processing based upon an own ID code and an ID code received from other equipment has been carried out, said method comprising the steps of:

reading out said own ID code from a storage means;
receiving an ID code transmitted by other equipment;
calculating a 1st value by application of a predetermined sub-method to said ID read out from said storage means and said ID received from said other equipment;
generating key information;
carrying out said predetermined processing based upon said 1st value on said key information; and
transmitting said key information completing said predetermined processing.

42. A recording medium for recording a program prescribing a data transmitting method according to claim 41 wherein, in accordance with said predetermined sub-method, a predetermined function is applied a predetermined number of times.

43. A recording medium for recording a program prescribing a data receiving method for decrypting data received from other equipment by decryption based on an own key and information received from said other equipment wherein said method comprises the steps of:

reading out said own key from a storage means;
receiving data and information from other equipment; and
decrypting said data by using said key and said information as a base.

44. A recording medium for recording a program prescribing a data receiving method according to claim 43 wherein a judgment as to whether or not said method' own key satisfies a predetermined condition is formed and predetermined processing is carried out on said method' own key till said method' own key satisfies said predetermined condition.

FIG. 1

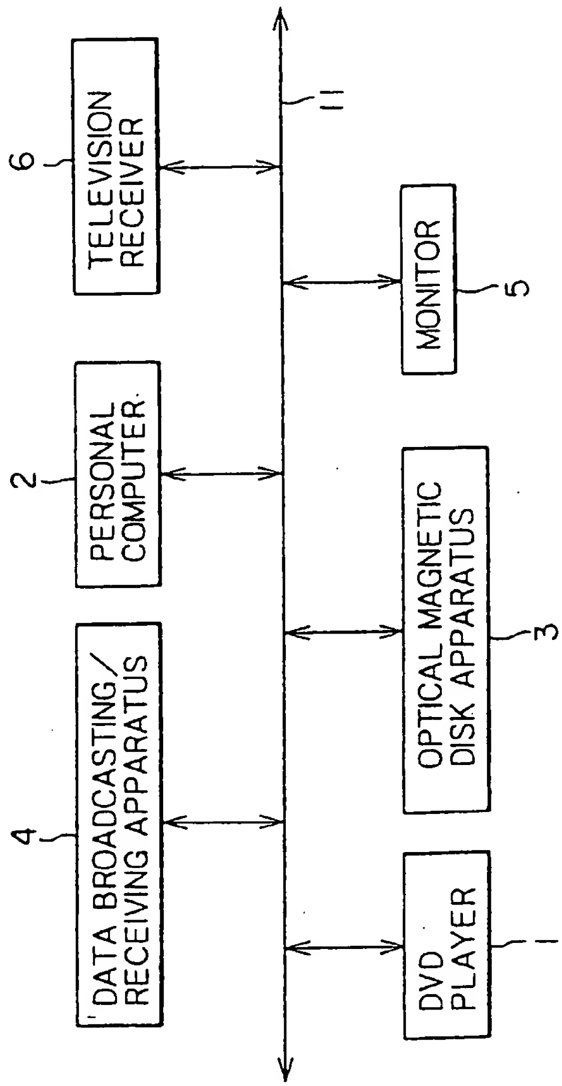

TELEVISION RECEIVER
6
PERSONAL COMPUTER
2
DATA BROADCASTING/ RECEIVING APPARATUS
4
11
MONITOR
5
OPTICAL MAGNETIC DISK APPARATUS
3
DVD PLAYER
1

FIG. 2

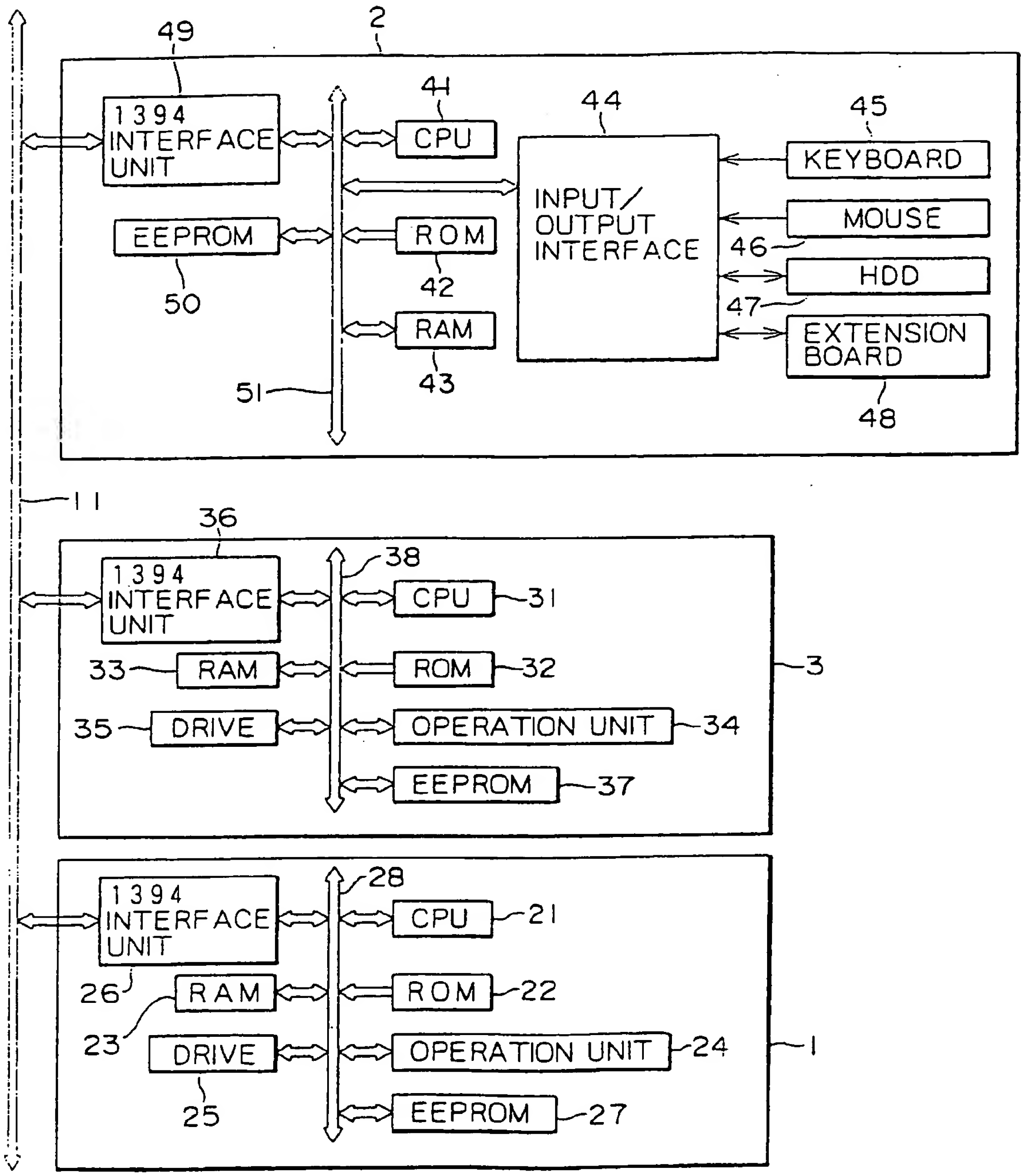

2
49
1394 INTERFACE UNIT
EEPROM
50
41
CPU
ROM
42
RAM
43
51
44
INPUT/ OUTPUT INTERFACE
45
KEYBOARD
MOUSE
46
HDD
47
EXTENSION BOARD
48
11
36
1394 INTERFACE UNIT
33
RAM
35
DRIVE
38
CPU
31
ROM
32
OPERATION UNIT
34
EEPROM
37
3
1394 INTERFACE UNIT
26
RAM
23
DRIVE
25
28
CPU
21
ROM
22
OPERATION UNIT
24
EEPROM
27
1

FIG. 3

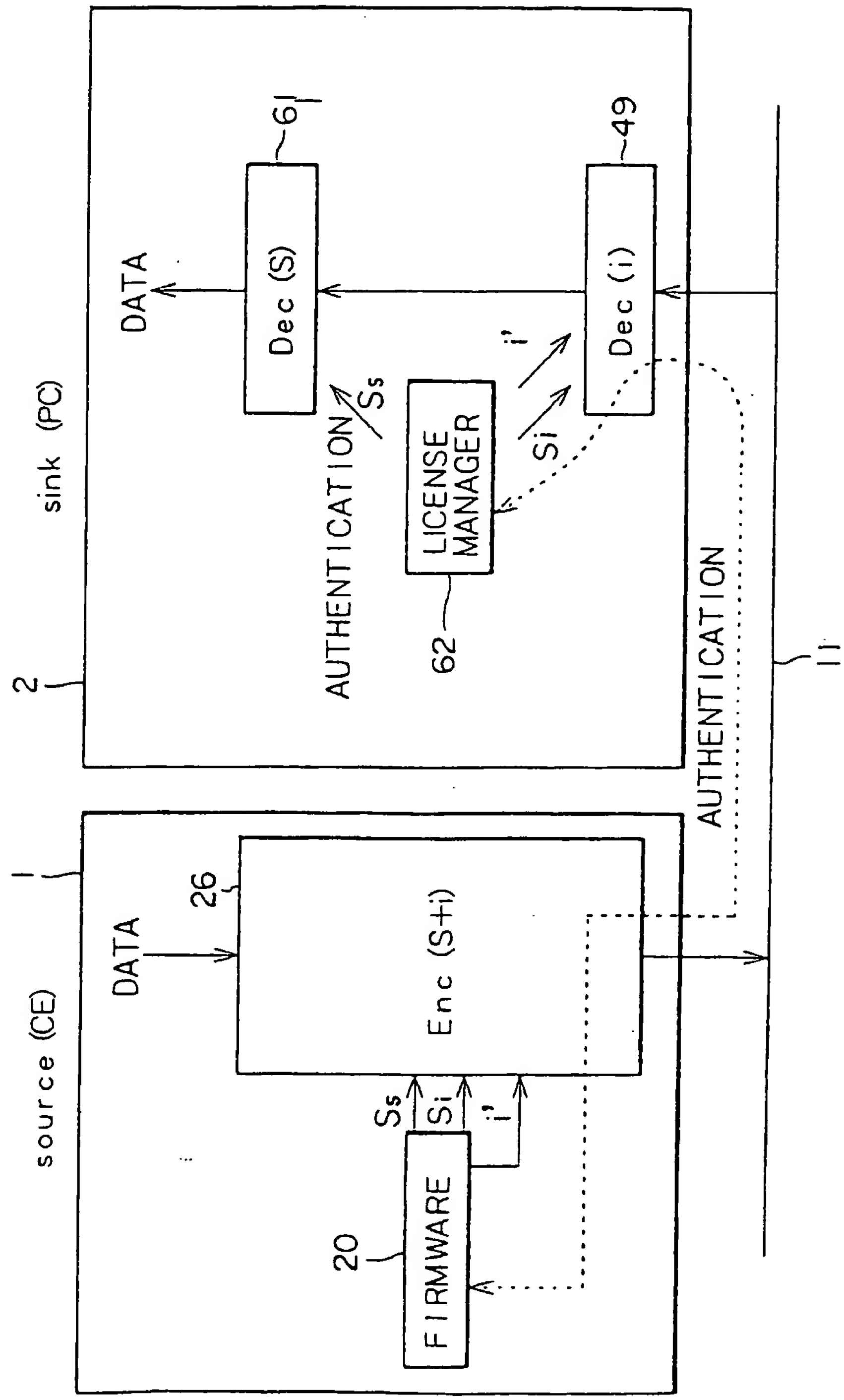

source (CE)
1
DATA
26
Enc (S+i)
Ss
Si
i'
20
FIRMWARE
sink (PC)
2
DATA
Dec (S)
61
AUTHENTICATION
Ss
LICENSE MANAGER
62
Si
i'
Dec (i)
49
AUTHENTICATION
11

FIG.4

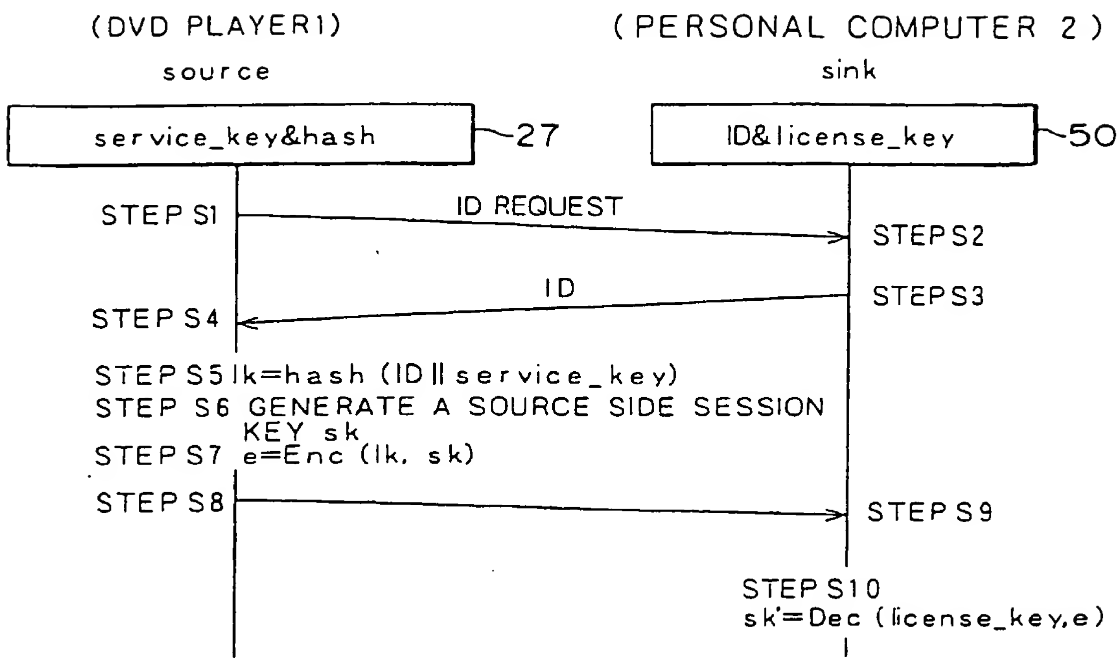

(DVD PLAYER1)
source
(PERSONAL COMPUTER 2)
sink
service_key&hash
27
ID&license_key
50
STEP S1
ID REQUEST
STEP S2
ID
STEP S3
STEP S4
STEP S5 lk=hash (ID|| service_key)
STEP S6 GENERATE A SOURCE SIDE SESSION KEY sk
STEP S7 e=Enc (lk, sk)
STEP S8
STEP S9
STEP S10
sk'=Dec (license_key,e)

FIG.5

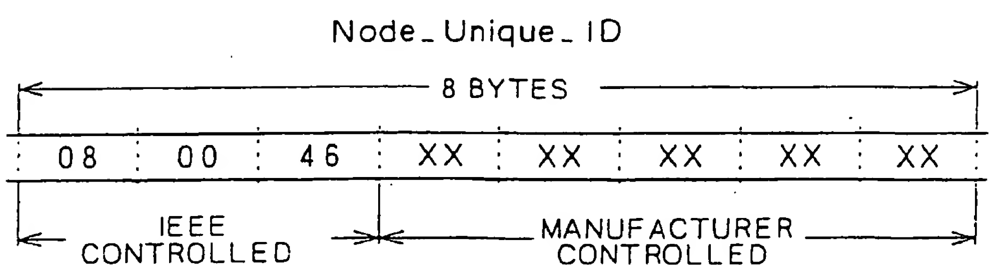

Node_Unique_ID
8 BYTES
08
00
46
XX
XX
XX
XX
XX
IEEE CONTROLLED
MANUFACTURER CONTROLLED

FIG. 6

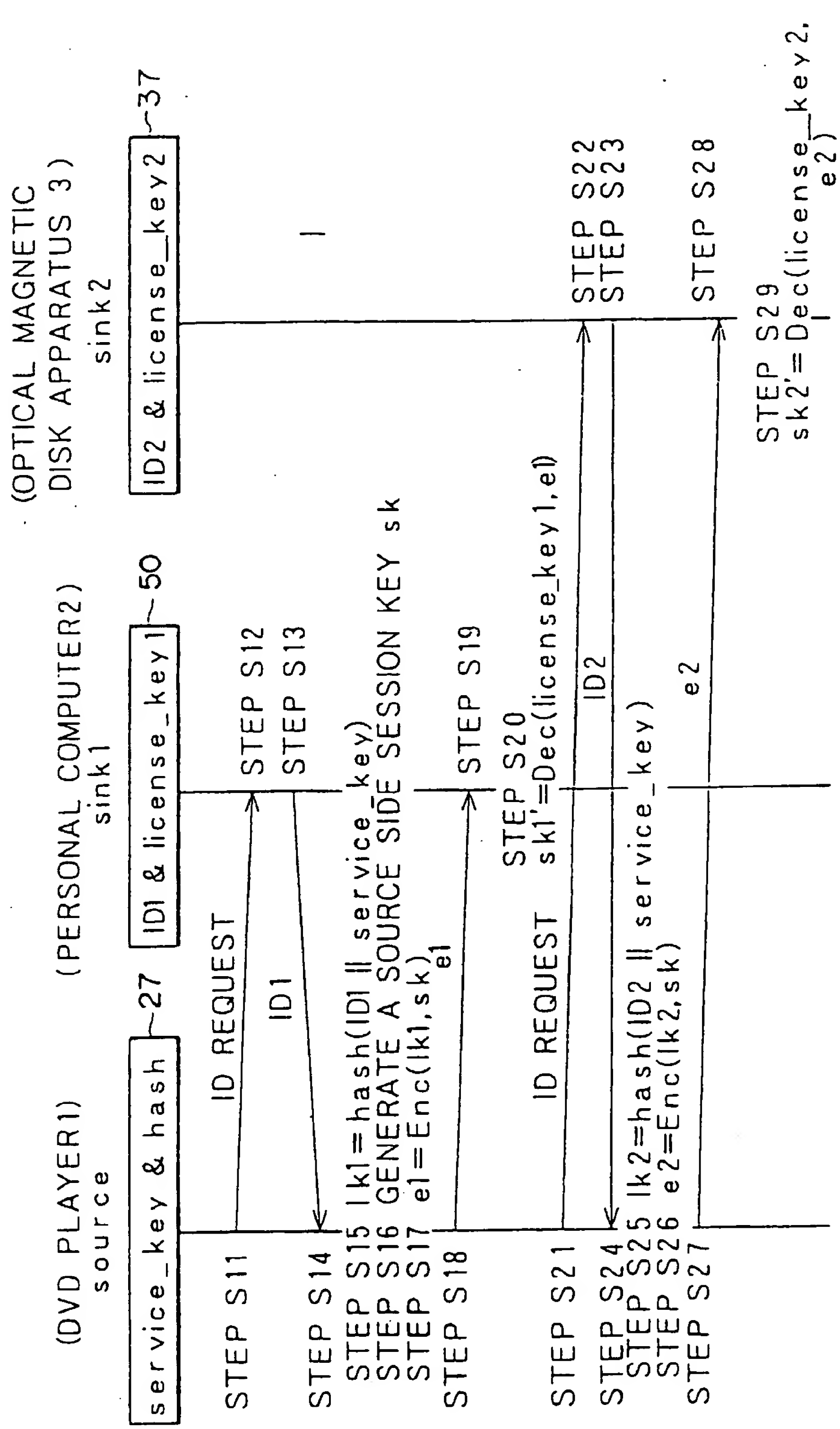

(DVD PLAYER1)
source
service_key & hash
27
(PERSONAL COMPUTER2)
sink1
ID1 & license_key1
50
(OPTICAL MAGNETIC DISK APPARATUS 3)
sink2
ID2 & license_key2
37
STEP S11
ID REQUEST
STEP S12
ID1
STEP S13
STEP S14
STEP S15 lk1=hash(ID1 ‖ service_key)
STEP S16 GENERATE A SOURCE SIDE SESSION KEY sk
STEP S17 e1=Enc(lk1,sk)
e1
STEP S18
STEP S19
STEP S20
sk1'=Dec(license_key1,e1)
STEP S21
ID REQUEST
STEP S22
ID2
STEP S23
STEP S24
STEP S25 lk2=hash(ID2 ‖ service_key)
STEP S26 e2=Enc(lk2,sk)
STEP S27
e2
STEP S28
STEP S29
sk2'=Dec(license_key2, e2)

FIG. 7

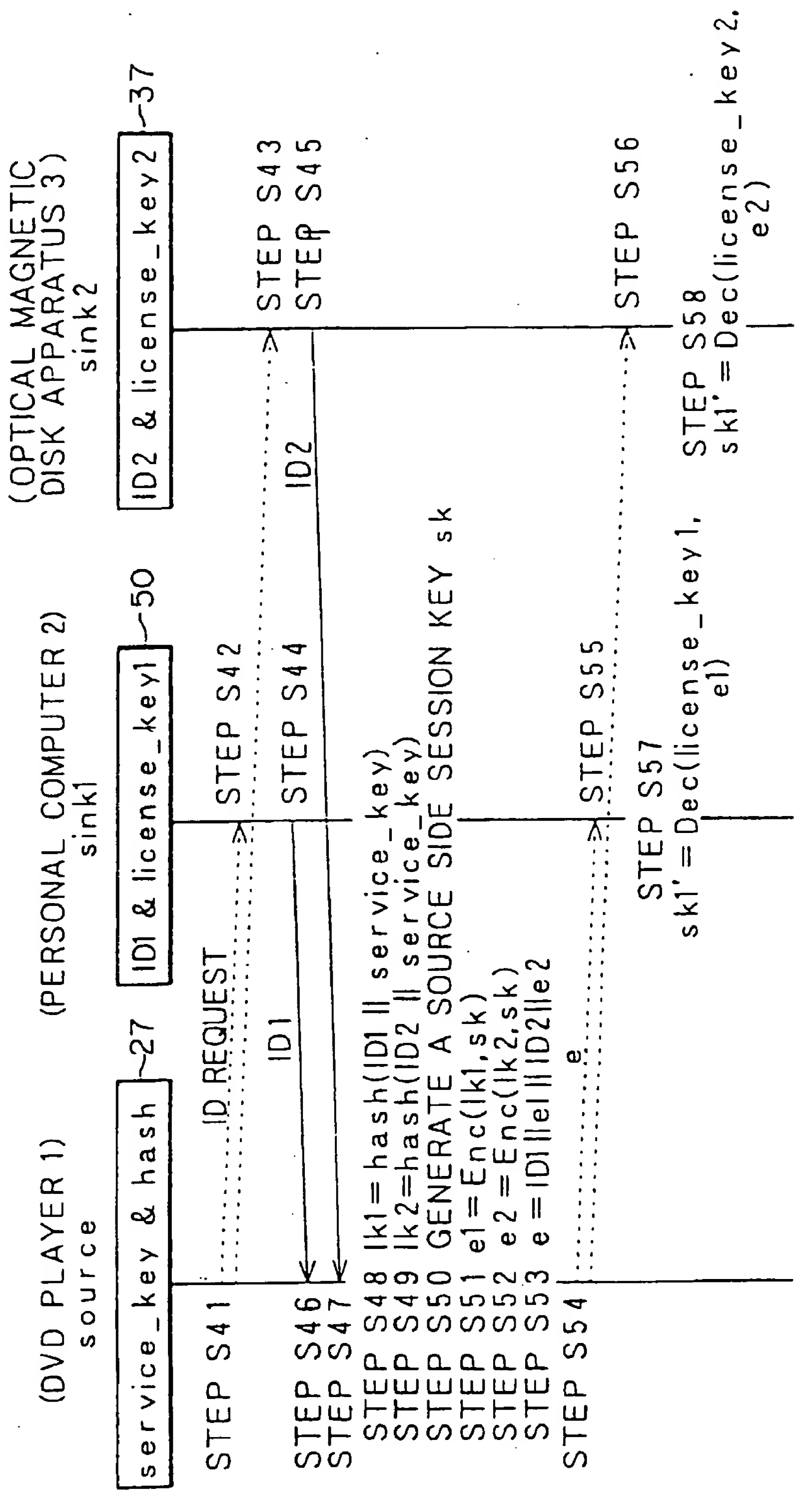

(DVD PLAYER 1)
source
service_key & hash
27
(PERSONAL COMPUTER 2)
sink1
ID1 & license_key1
50
(OPTICAL MAGNETIC DISK APPARATUS 3)
sink2
ID2 & license_key2
37
STEP S41
ID REQUEST
STEP S42
STEP S43
ID1
STEP S44
ID2
STEP S45
STEP S46
STEP S47
STEP S48 lk1=hash(ID1 || service_key)
STEP S49 lk2=hash(ID2 || service_key)
STEP S50 GENERATE A SOURCE SIDE SESSION KEY sk
STEP S51 e1=Enc(lk1,sk)
STEP S52 e2=Enc(lk2,sk)
STEP S53 e=ID1||e1||ID2||e2
STEP S54
e
STEP S55
STEP S56
STEP S57
sk1'=Dec(license_key1, e1)
STEP S58
sk1'=Dec(license_key2, e2)

FIG. 8

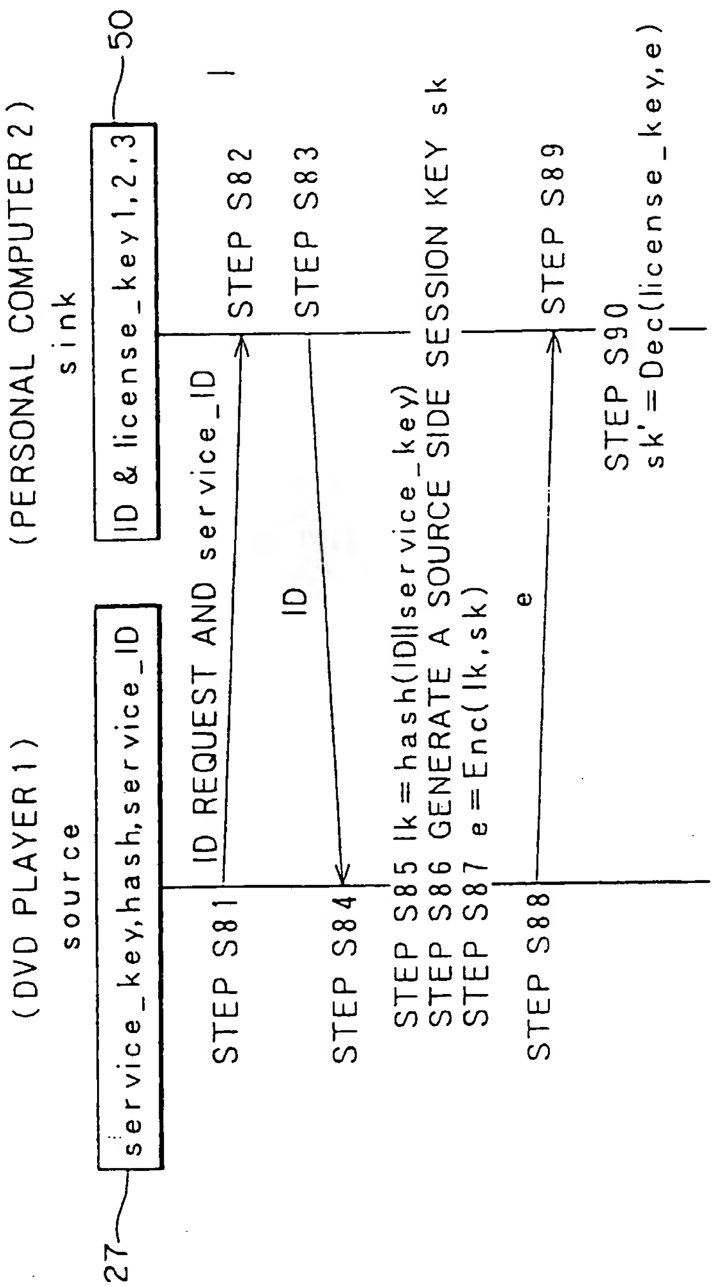

(DVD PLAYER 1)
source
service_key,hash,service_ID
27
(PERSONAL COMPUTER 2)
sink
ID & license_key1,2,3
50
STEP S81
ID REQUEST AND service_ID
STEP S82
STEP S83
ID
STEP S84
STEP S85 lk = hash(ID||service_key)
STEP S86 GENERATE A SOURCE SIDE SESSION KEY sk
STEP S87 e = Enc(lk,sk)
e
STEP S88
STEP S89
STEP S90
sk' = Dec(license_key,e)

FIG. 9

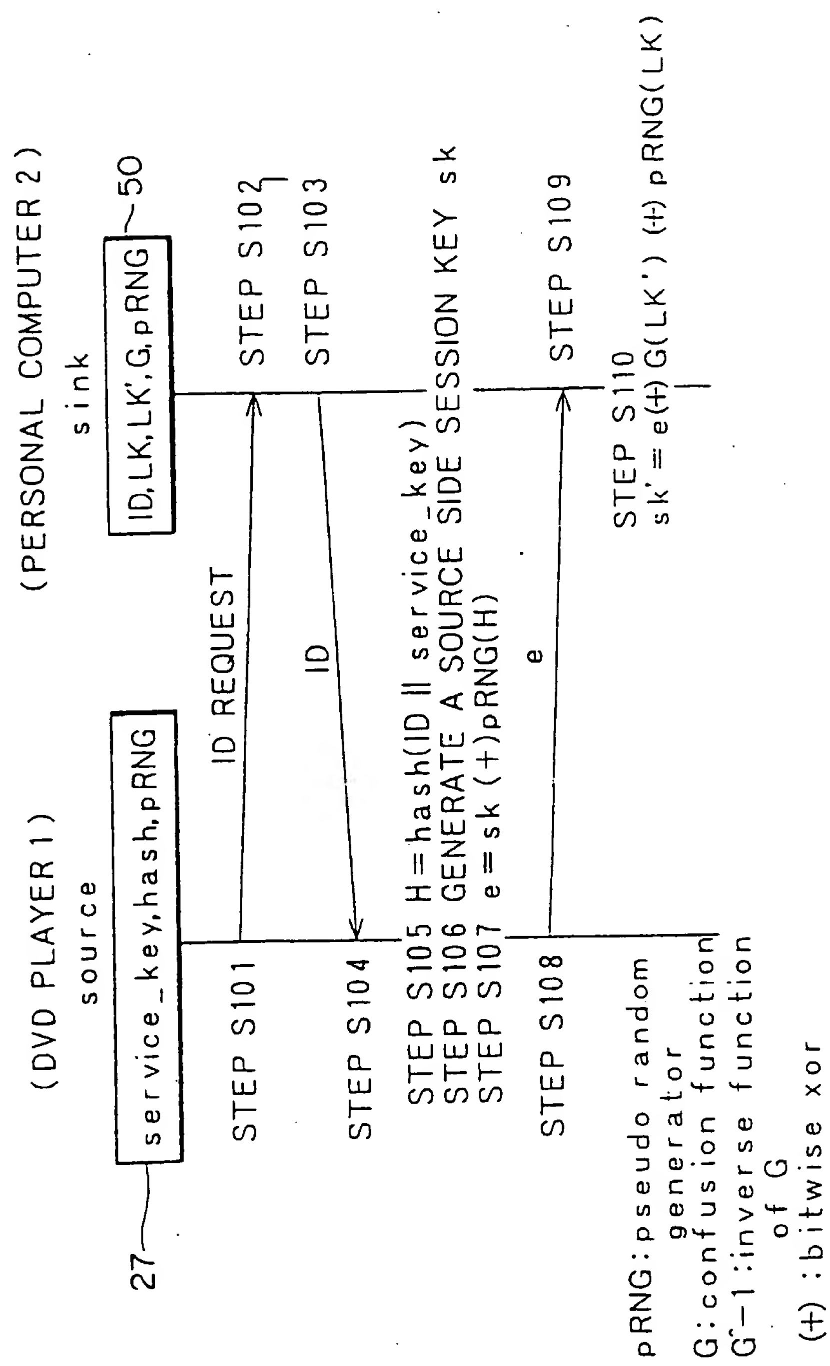

(DVD PLAYER 1)
source
service_key,hash,pRNG
27
(PERSONAL COMPUTER 2)
sink
ID,LK,LK',G,pRNG
50
STEP S101
ID REQUEST
STEP S102
STEP S103
ID
STEP S104
STEP S105 H=hash(ID || service_key)
STEP S106 GENERATE A SOURCE SIDE SESSION KEY sk
STEP S107 e = sk (+) pRNG(H)
e
STEP S108
STEP S109
STEP S110
sk' = e(+) G(LK') (+) pRNG(LK)
pRNG: pseudo random generator
G: confusion function
G^-1: inverse function of G
(+) : bitwise xor

FIG. 10

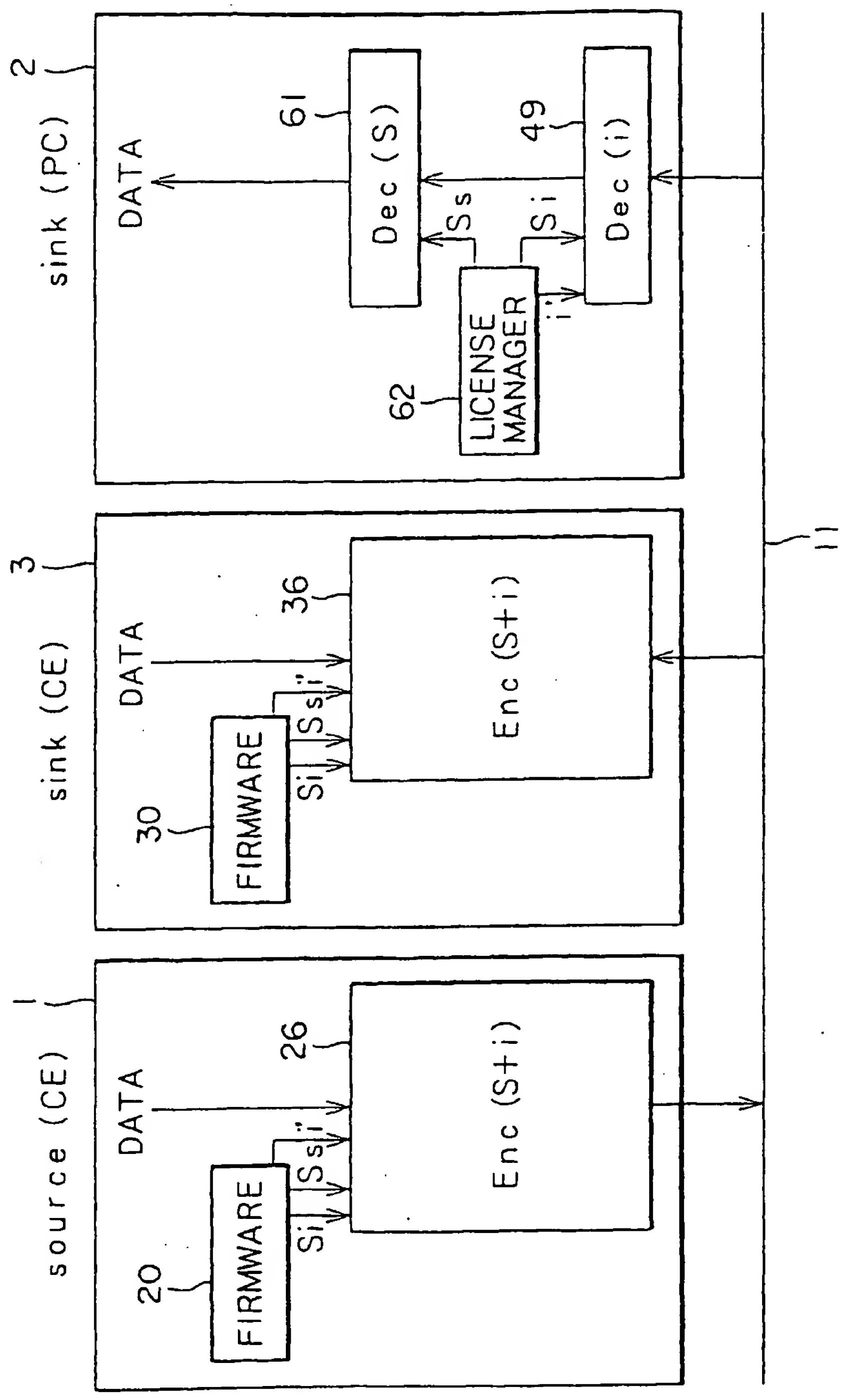

source (CE)
1
DATA
FIRMWARE
20
Si
Ssi'
26
Enc (S+i)
sink (CE)
3
DATA
FIRMWARE
30
Si
Ssi'
36
Enc (S+i)
sink (PC)
2
DATA
Dec (S)
61
Ss
Si
Dec (i)
49
LICENSE MANAGER
62
i'
11

FIG. 11

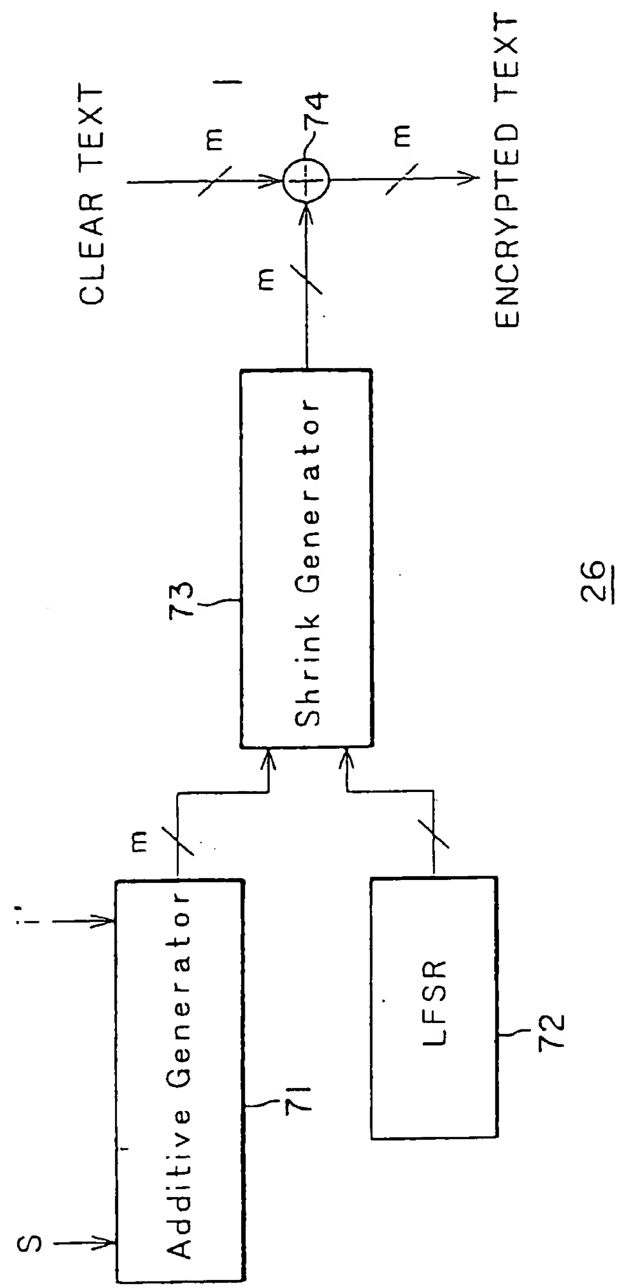

CLEAR TEXT
ENCRYPTED TEXT
m
m
m
m
74
73
Shrink Generator
26
i'
S
Additive Generator
71
LFSR
72

FIG. 12

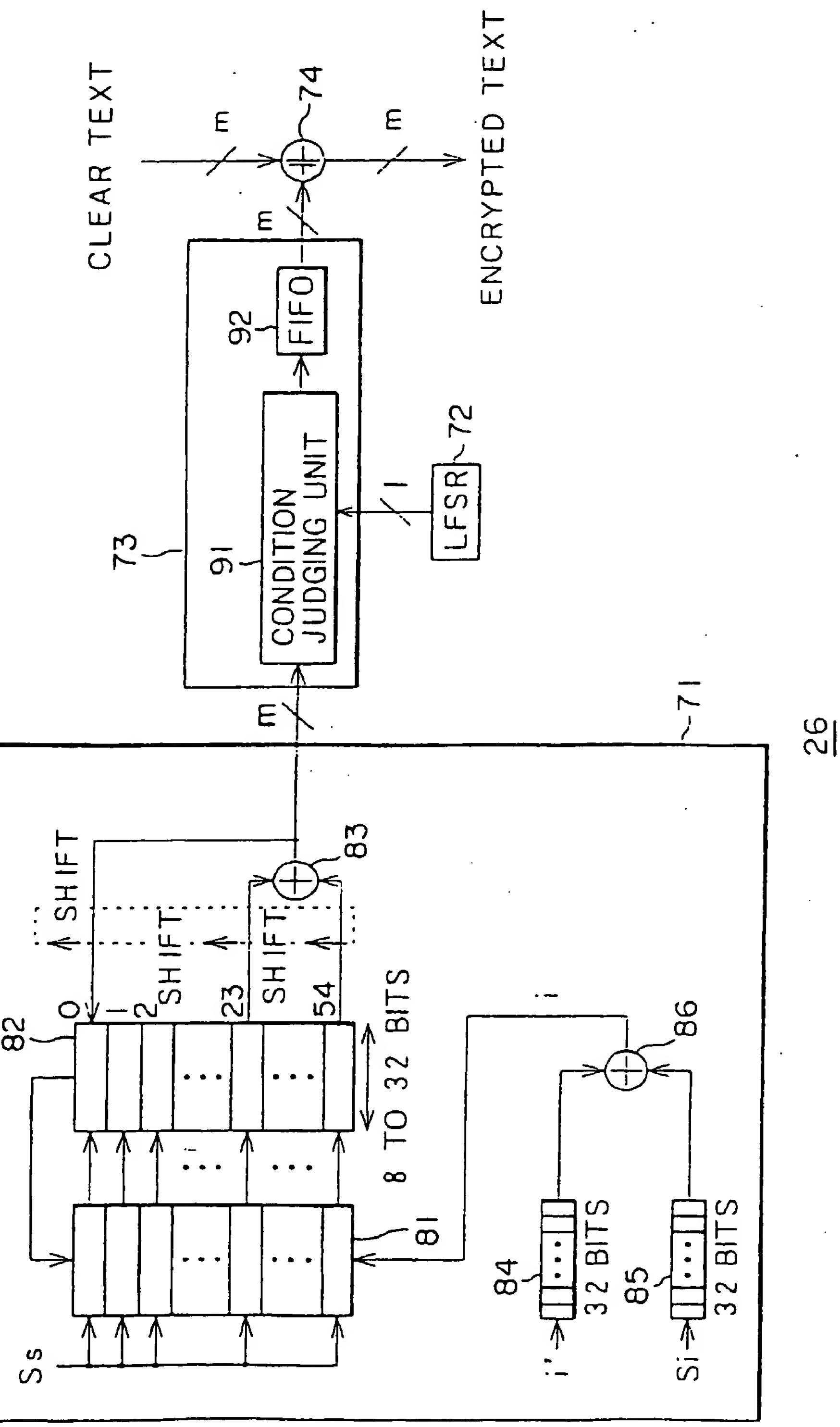

CLEAR TEXT
ENCRYPTED TEXT
m
74
73
91
CONDITION JUDGING UNIT
92
FIFO
LFSR
72
l
71
26
82
83
SHIFT
0
1
2
23
54
8 TO 32 BITS
81
86
84
32 BITS
85
32 BITS
Ss
i'
Si

FIG. 13

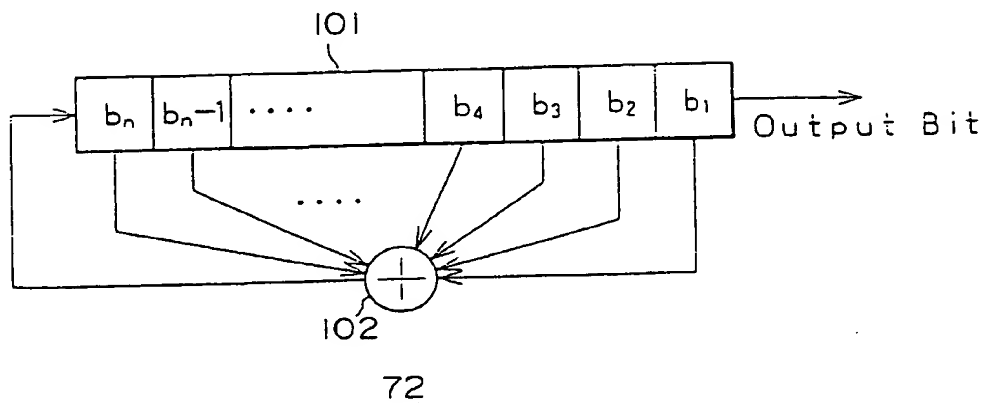

101
b_n
b_n-1
....
b_4
b_3
b_2
b_1
Output Bit
....
102
72

FIG. 14

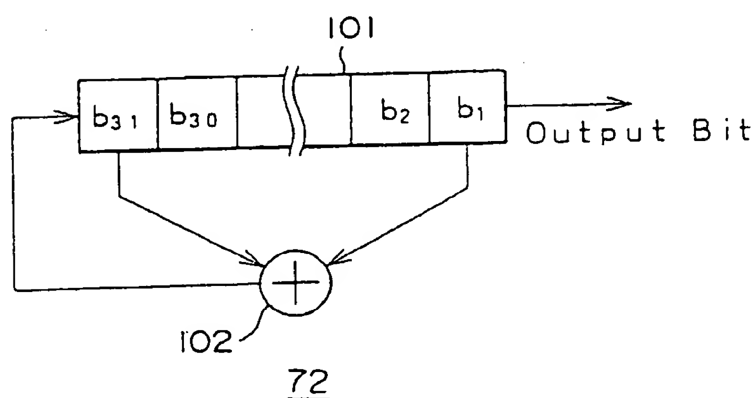
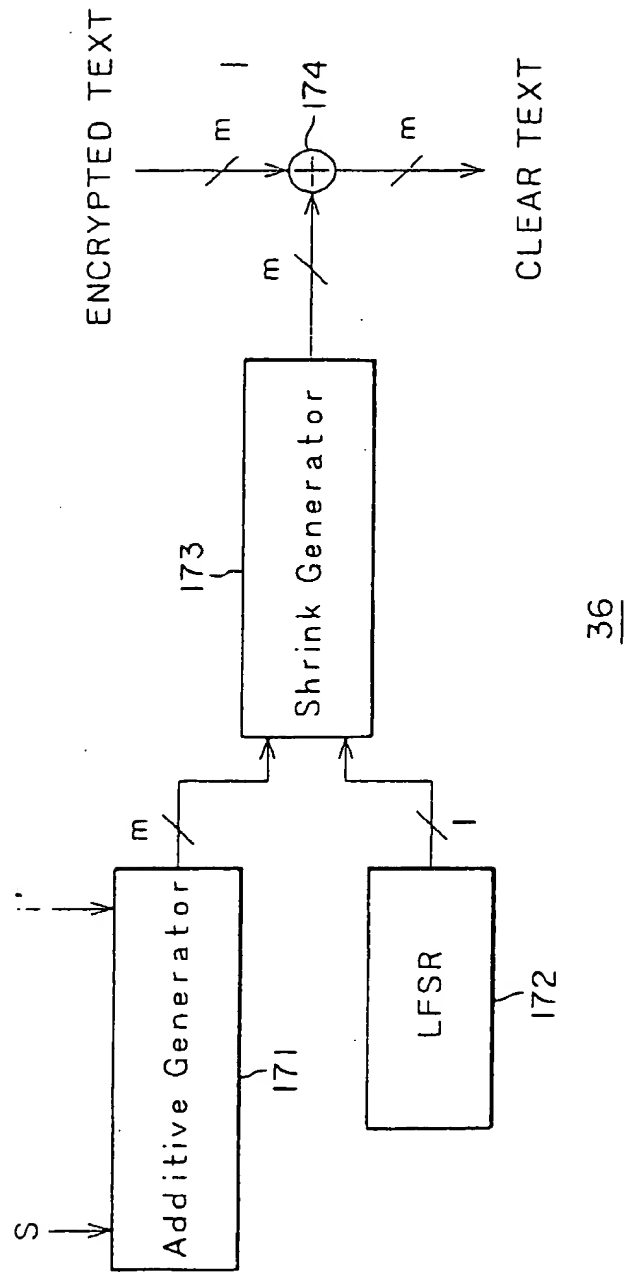

101
b_{31}
b_{30}
b_2
b_1
Output Bit
102
72

FIG. 15


ENCRYPTED TEXT
CLEAR TEXT
174
m
m
m
l
Shrink Generator
173
m
l
S
i
Additive Generator
171
LFSR
172
36

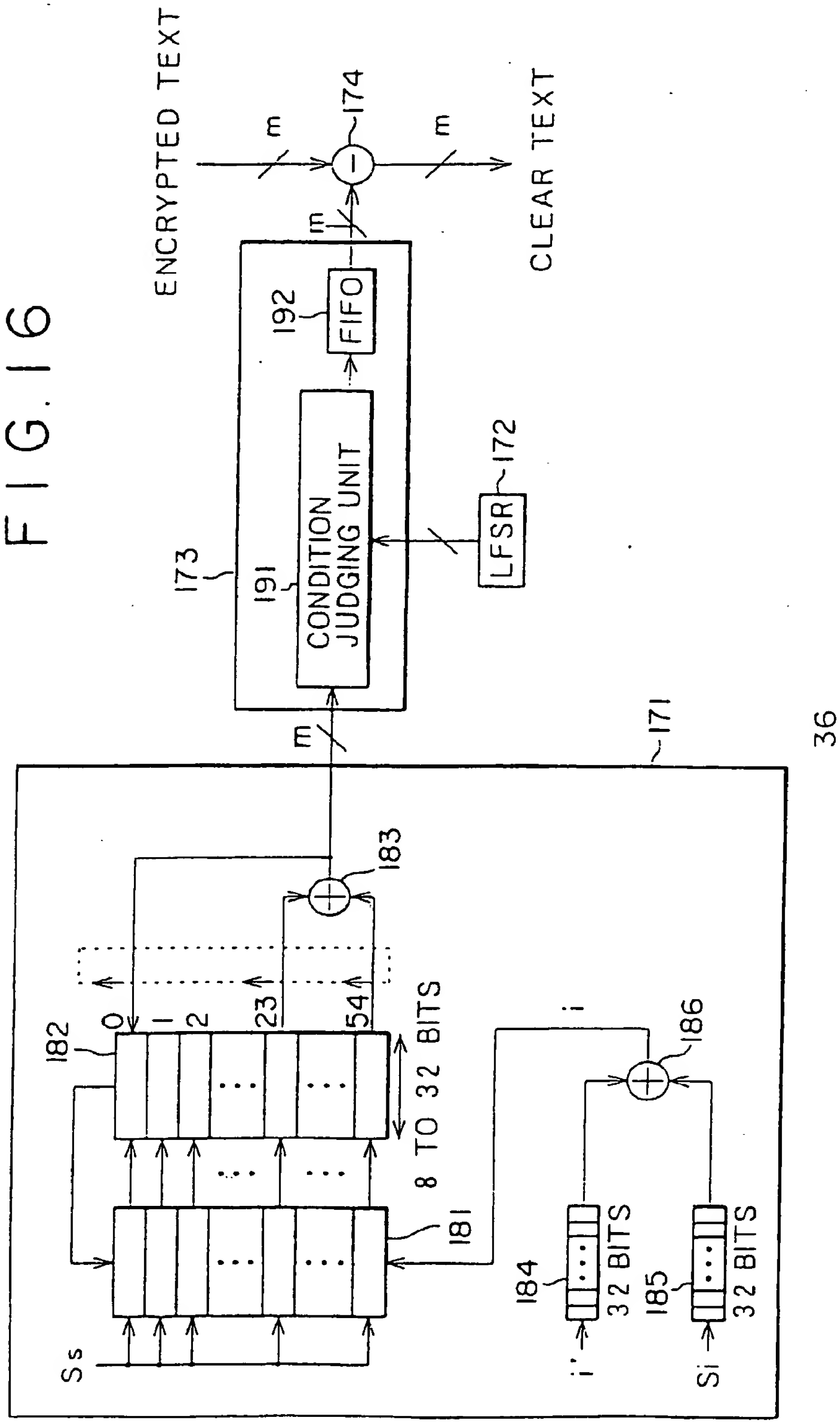

ENCRYPTED TEXT
CLEAR TEXT
m
174
192
FIFO
CONDITION JUDGING UNIT
191
173
LFSR
172
171
36
183
182
0
1
2
23
54
8 TO 32 BITS
181
i
186
184
32 BITS
185
32 BITS
Ss
i'
Si

FIG. 16

FIG. 17

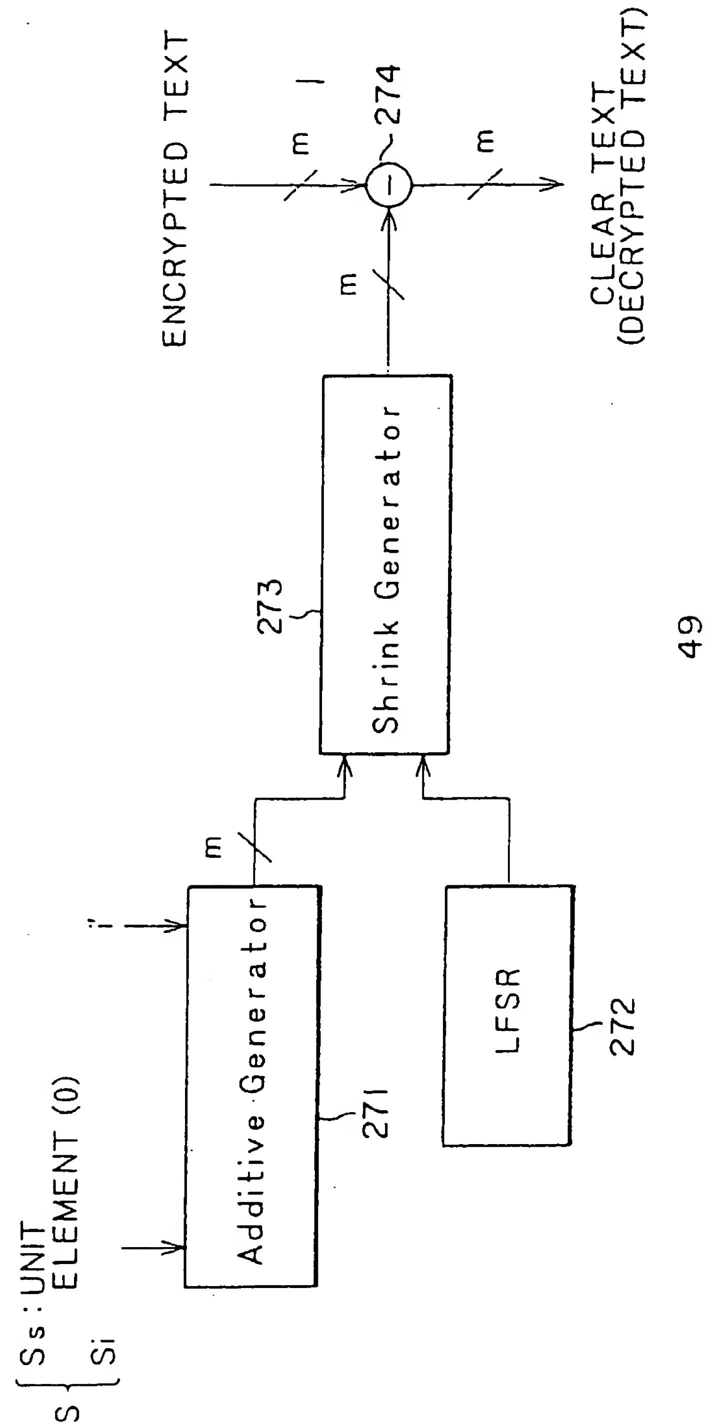

S { Ss : UNIT ELEMENT (0)
Si
i
Additive Generator
271
LFSR
272
m
Shrink Generator
273
m
ENCRYPTED TEXT
m
274
m
CLEAR TEXT (DECRYPTED TEXT)
49

FIG. 18

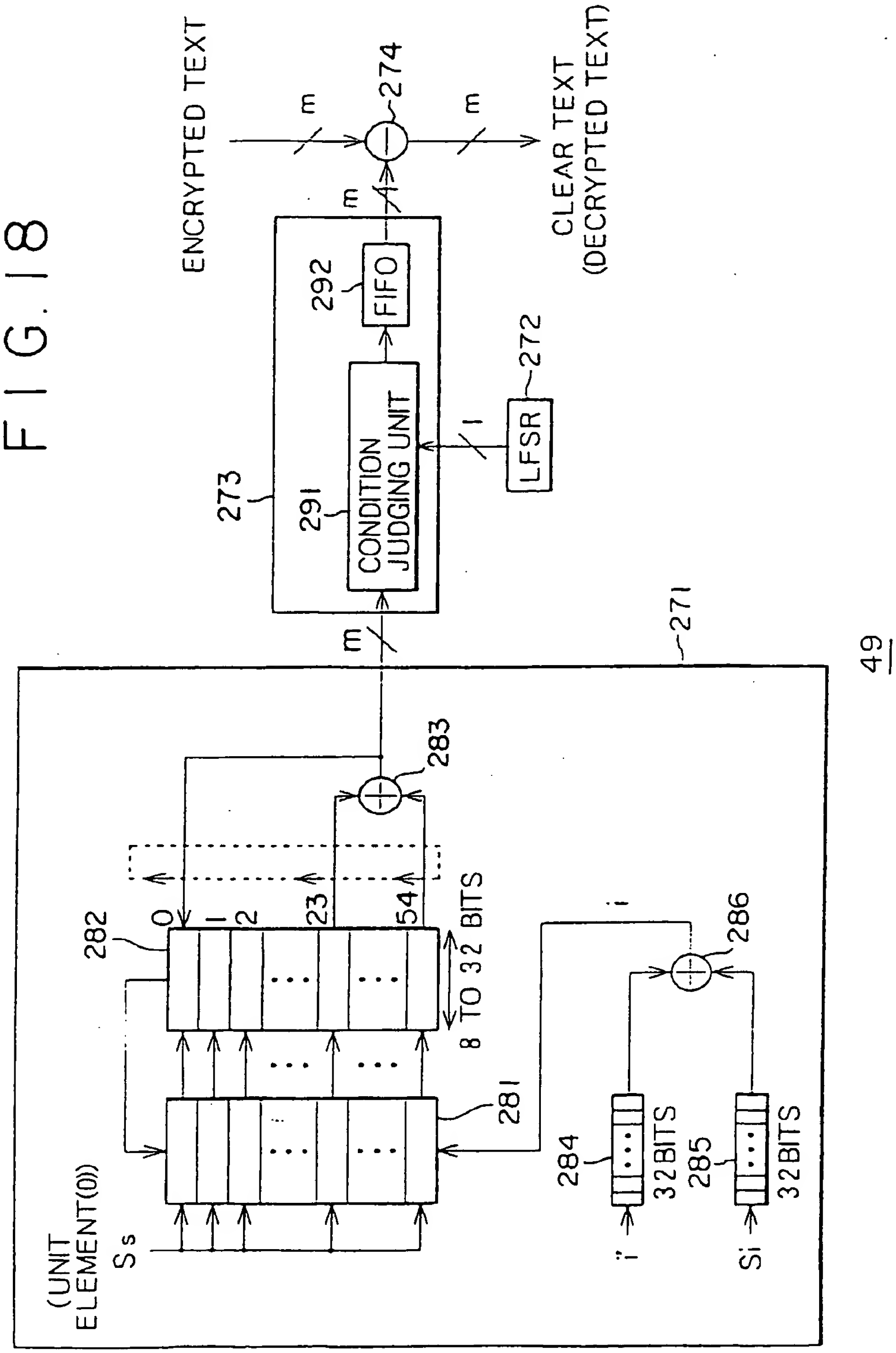
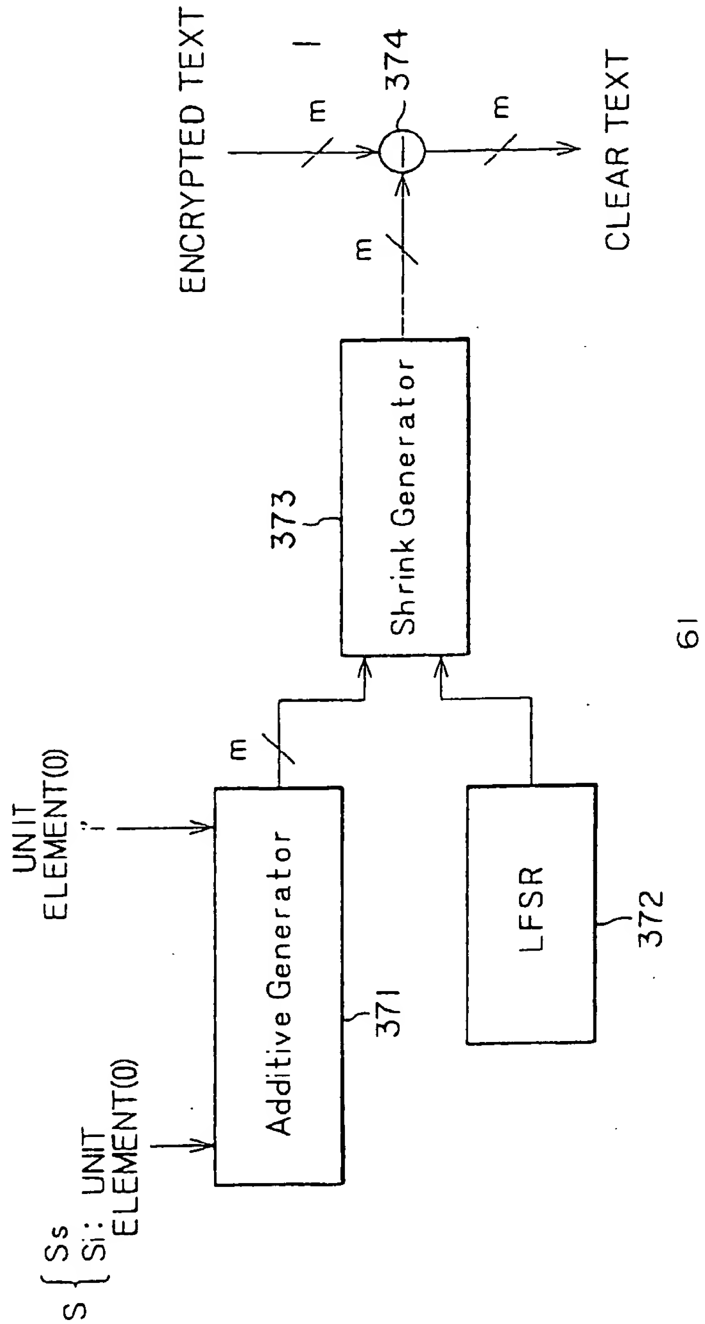
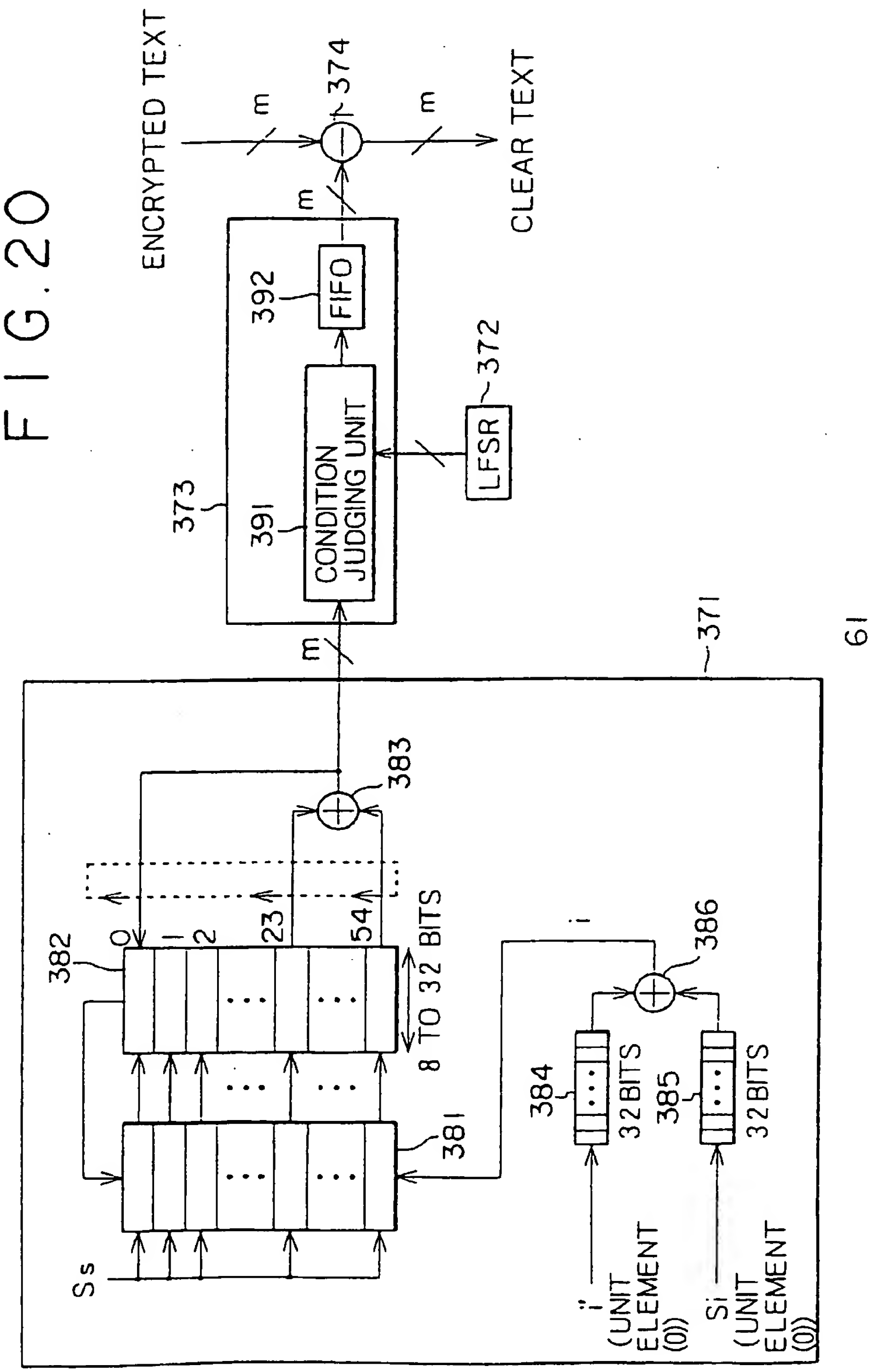
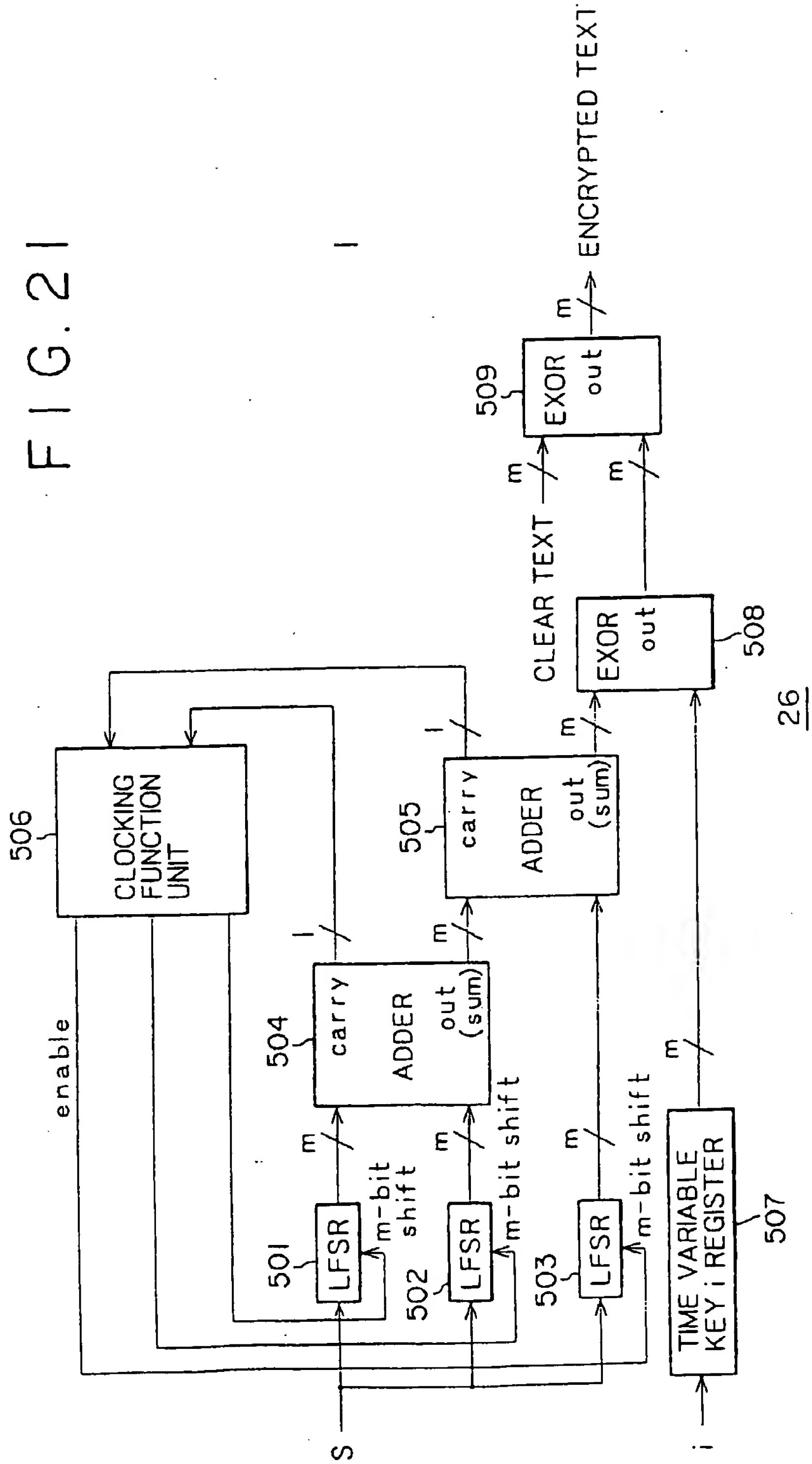

ENCRYPTED TEXT
m
274
m
CLEAR TEXT
(DECRYPTED TEXT)
m
292
FIFO
273
291
CONDITION
JUDGING UNIT
l
LFSR
272
m
271
49
283
282
0
1
2
23
54
8 TO 32 BITS
286
i
281
284
32 BITS
285
32 BITS
i
Si
(UNIT
ELEMENT(0))
Ss

FIG.19


ENCRYPTED TEXT
m
374
m
CLEAR TEXT
m
373
Shrink Generator
61
m
UNIT ELEMENT(0)
Additive Generator
371
LFSR
372
S
Ss
Si: UNIT ELEMENT(0)

F I G. 20

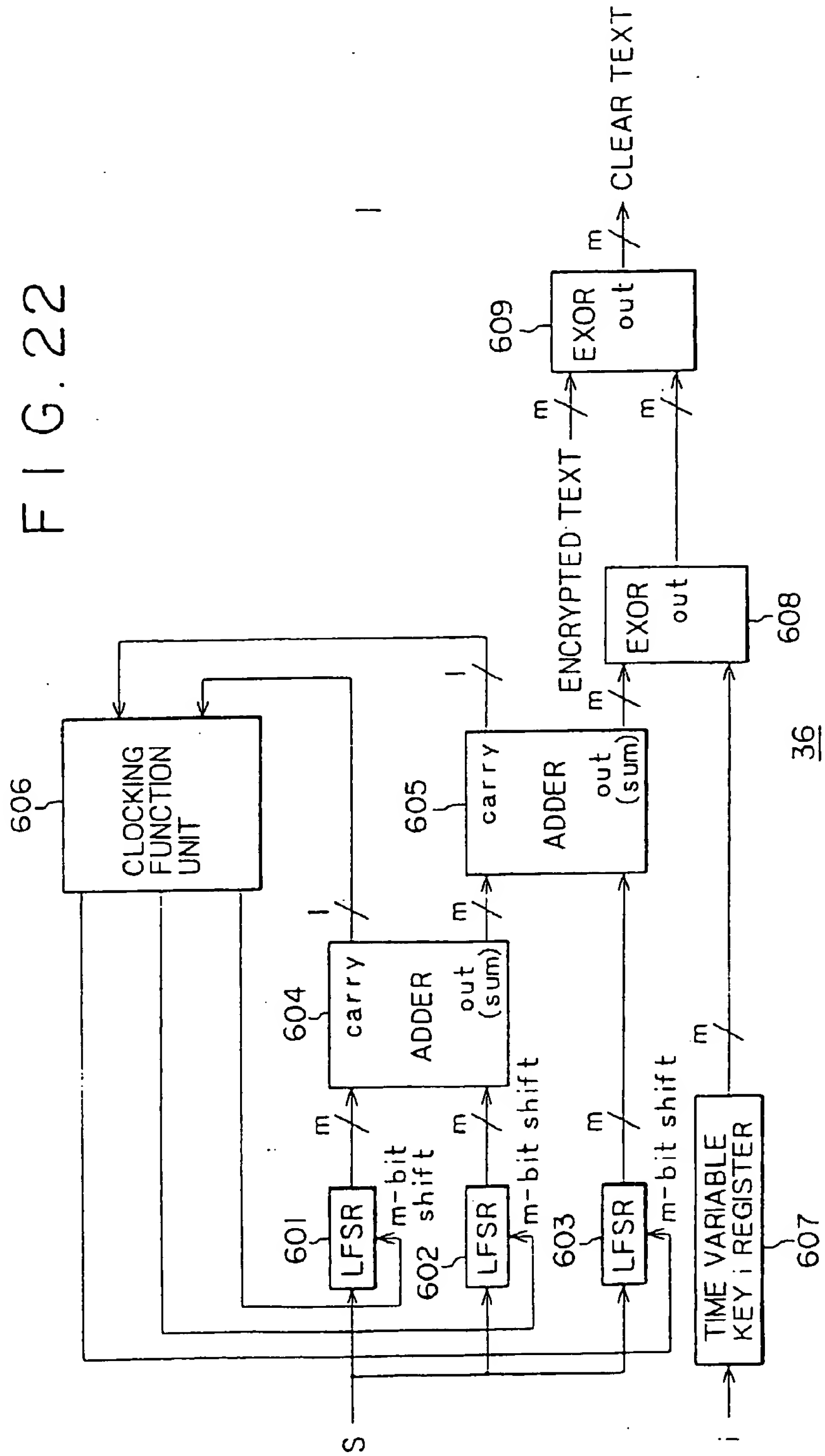

ENCRYPTED TEXT
CLEAR TEXT
m
374
373
392
FIFO
391
CONDITION JUDGING UNIT
LFSR
372
371
61
383
382
0
1
2
23
54
8 TO 32 BITS
381
384
385
32BITS
32BITS
386
i
Ss
i' (UNIT ELEMENT (0))
Si (UNIT ELEMENT (0))



CLOCKING FUNCTION UNIT
506
enable
LFSR
501
502
503
m-bit shift
ADDER
carry
out (sum)
504
505
m
I
S
TIME VARIABLE KEY i REGISTER
507
i
EXOR
out
508
509
CLEAR TEXT
ENCRYPTED TEXT
26

FIG. 21

FIG. 22



CLEAR TEXT
EXOR
out
609
ENCRYPTED TEXT
EXOR
out
608
CLOCKING FUNCTION UNIT
606
carry
ADDER
out (sum)
605
carry
ADDER
out (sum)
604
LFSR
601
m-bit shift
LFSR
602
m-bit shift
LFSR
603
m-bit shift
TIME VARIABLE KEY i REGISTER
607
S
i
m
l
36

FIG. 23

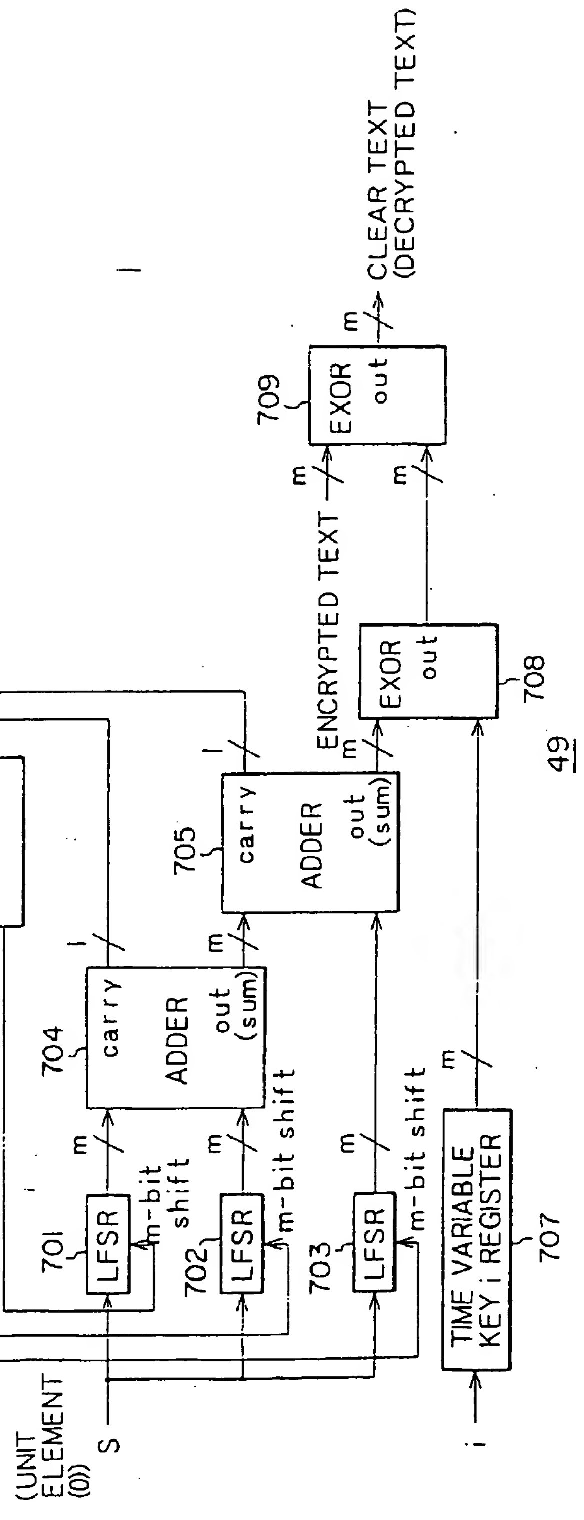

CLOCKING FUNCTION UNIT
706
LFSR
701
702
703
m-bit shift
ADDER
carry
out (sum)
704
705
m
l
ENCRYPTED TEXT
EXOR
out
708
709
CLEAR TEXT (DECRYPTED TEXT)
TIME VARIABLE KEY i REGISTER
707
S
(UNIT ELEMENT (0))
i
49

FIG. 24

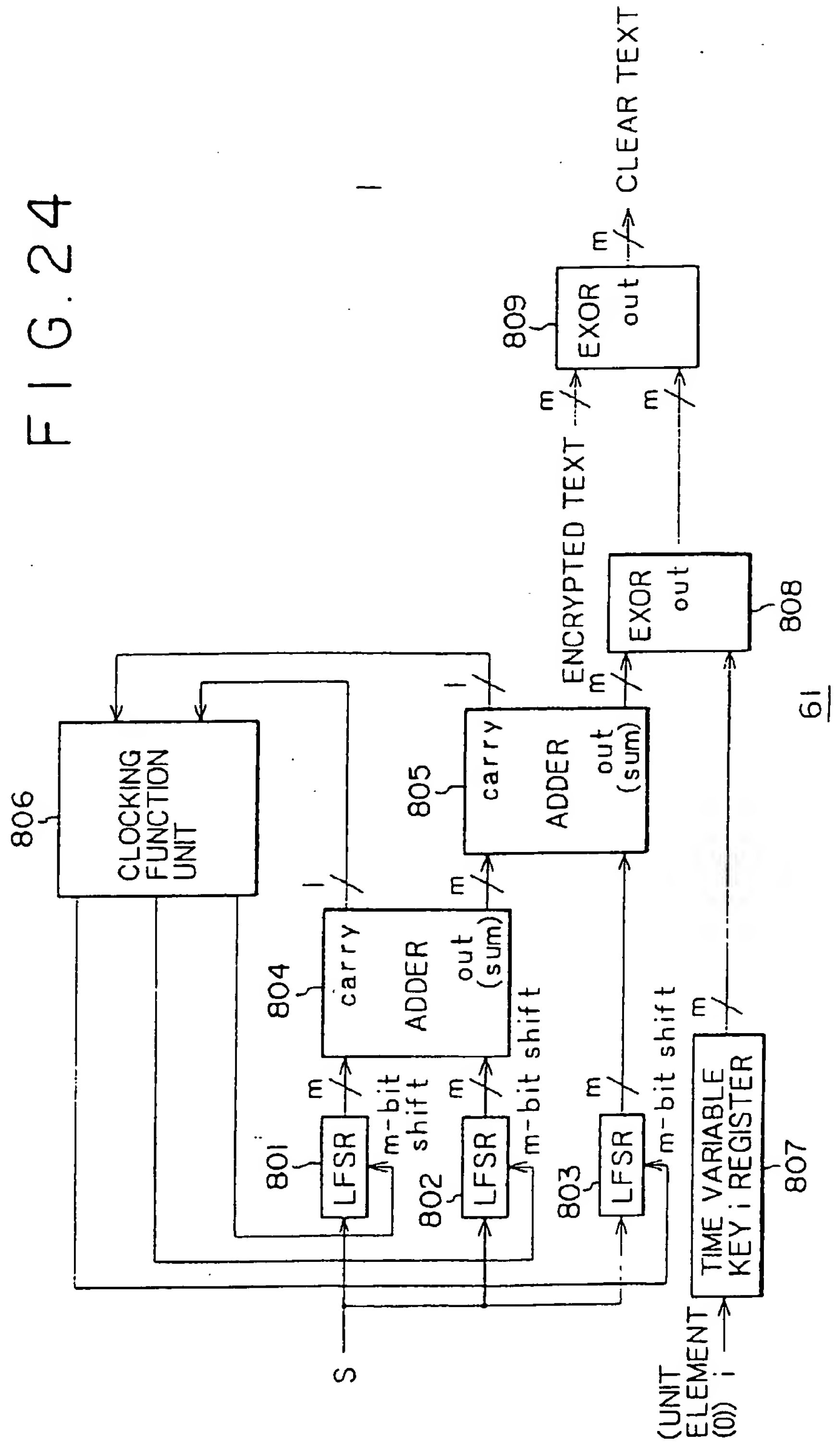

CLOCKING FUNCTION UNIT
806
LFSR
801
LFSR
802
LFSR
803
m-bit shift
m-bit shift
m-bit shift
S
carry
ADDER
out (sum)
804
carry
ADDER
out (sum)
805
EXOR
out
808
EXOR
out
809
TIME VARIABLE KEY i REGISTER
807
(UNIT ELEMENT (0)) i
ENCRYPTED TEXT
CLEAR TEXT
m
l
61

FIG. 25

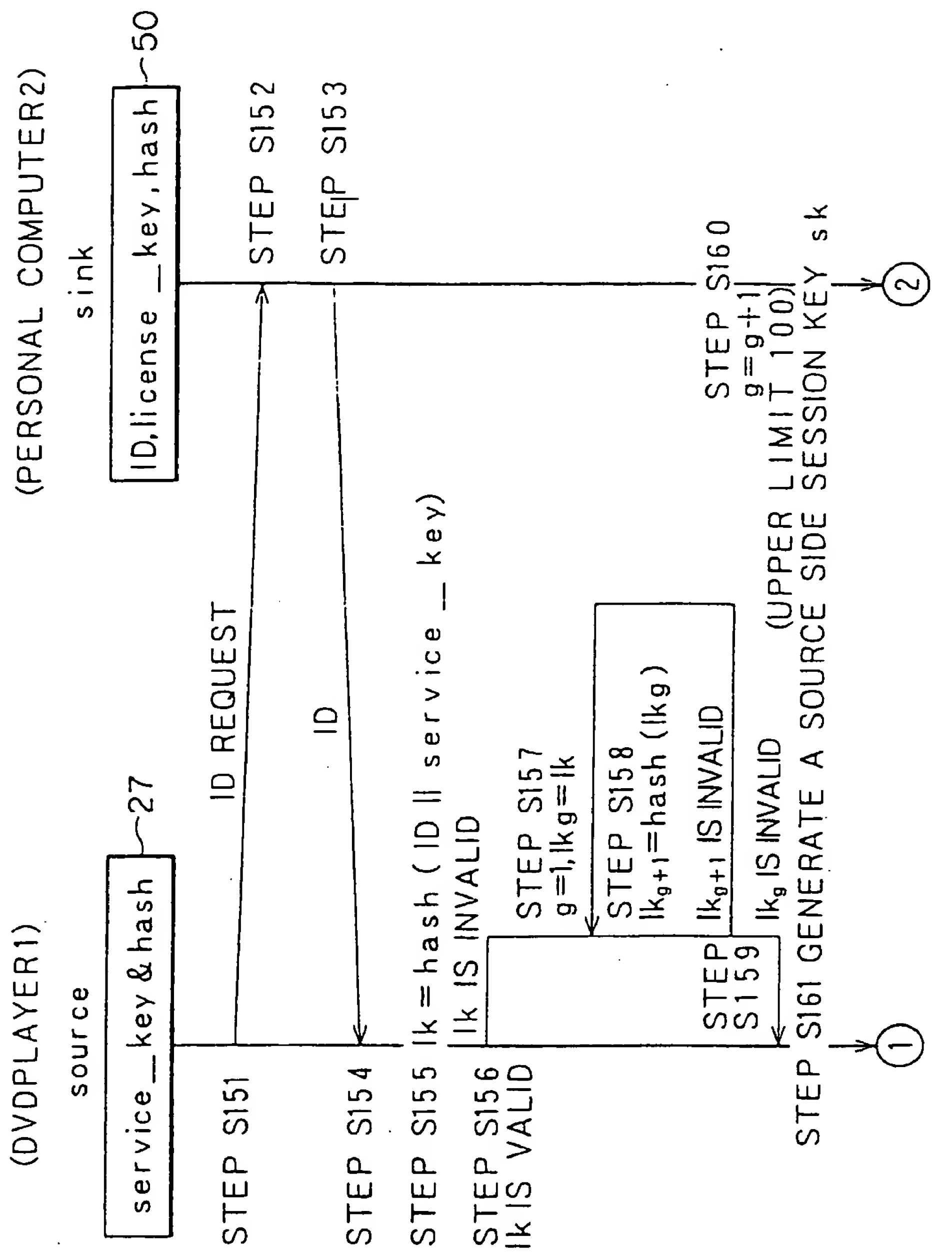

(DVDPLAYER1)
source
service__key&hash
27
(PERSONAL COMPUTER2)
sink
ID,license __key,hash
50
STEP S151
ID REQUEST
STEP S152
STEP S153
ID
STEP S154
STEP S155 lk=hash (ID || service __ key)
lk IS INVALID
STEP S156
lk IS VALID
STEP S157
g=1,lkg=lk
STEP S158
lkg+1=hash (lkg)
lkg+1 IS INVALID
STEP S159
lkg IS INVALID
STEP S160
g=g+1
(UPPER LIMIT 100)
STEP S161 GENERATE A SOURCE SIDE SESSION KEY sk
1
2

FIG. 26

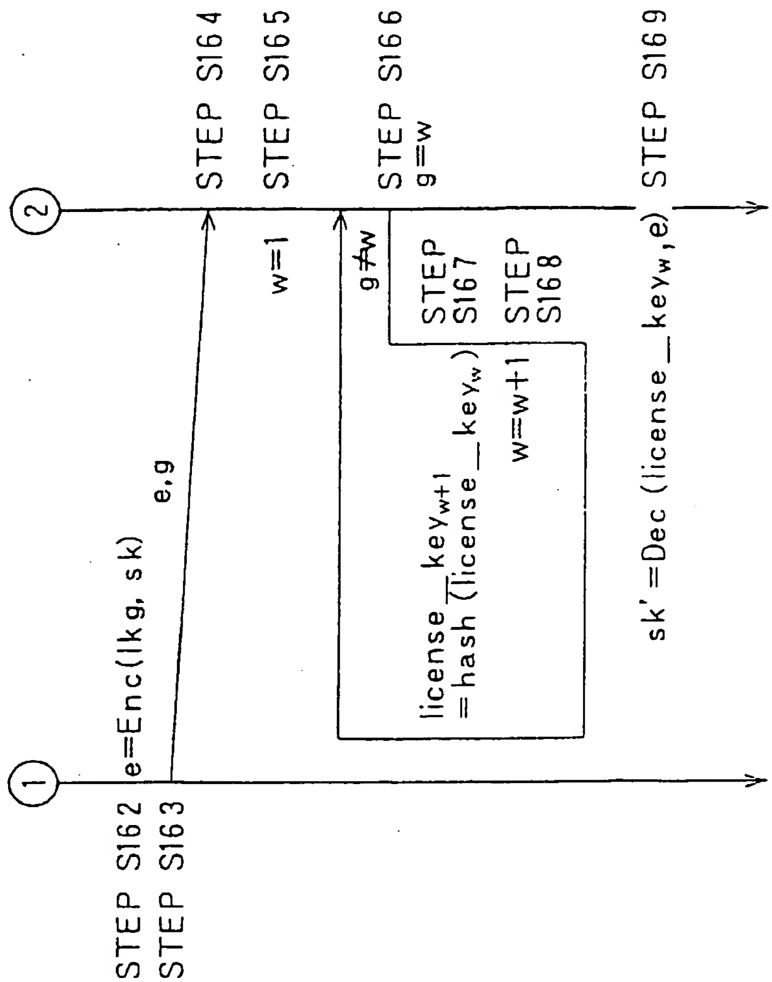

1
2
STEP S162
e=Enc(lkg, sk)
STEP S163
e,g
STEP S164
STEP S165
w=1
STEP S166
g=w
g≠w
STEP S167
license_key$_{w+1}$
=hash (license_key$_w$)
STEP S168
w=w+1
STEP S169
sk'=Dec (license_key$_w$,e)

FIG. 27

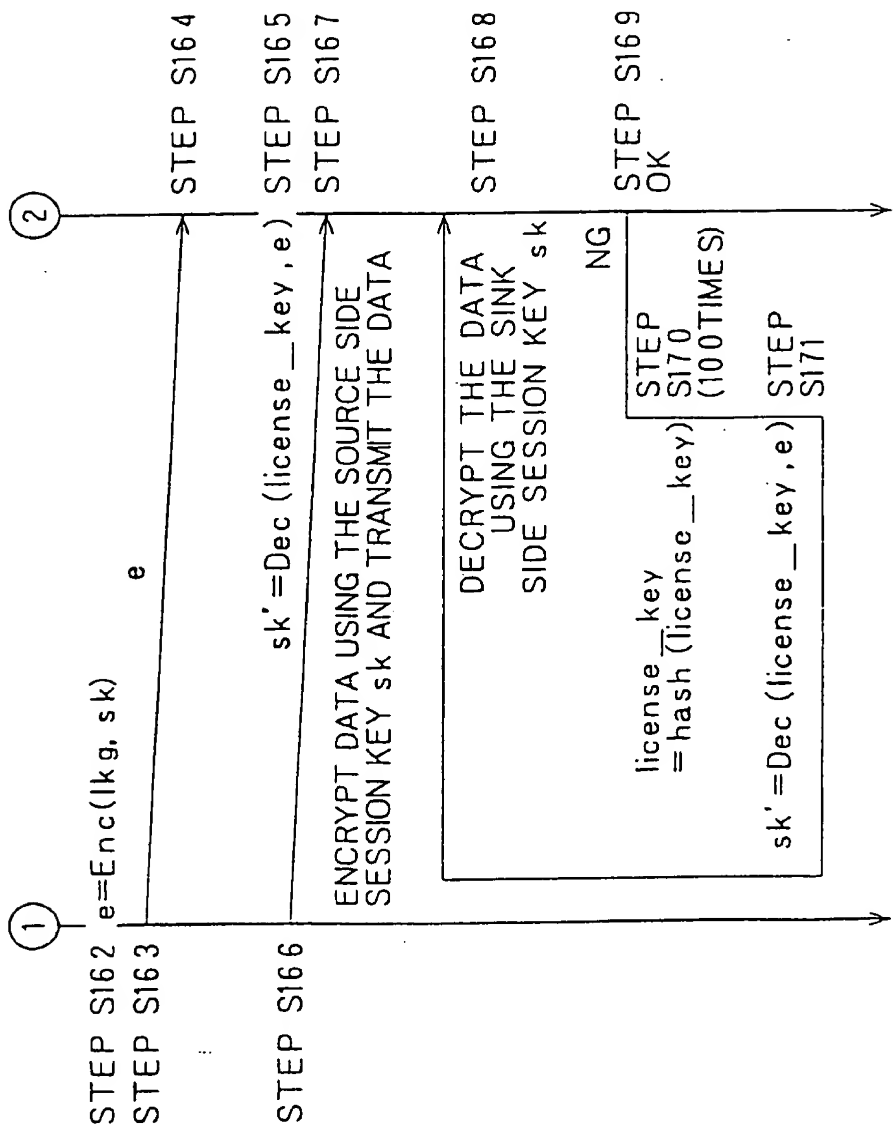

STEP S162
STEP S163
STEP S166
e=Enc(lkg, sk)
e
STEP S164
sk'=Dec (license__key, e)
STEP S165
STEP S167
ENCRYPT DATA USING THE SOURCE SIDE SESSION KEY sk AND TRANSMIT THE DATA
DECRYPT THE DATA USING THE SINK SIDE SESSION KEY sk
STEP S168
NG
STEP S169
OK
license__key =hash (license__key)
STEP S170 (100TIMES)
sk'=Dec (license__key, e)
STEP S171

FIG. 28

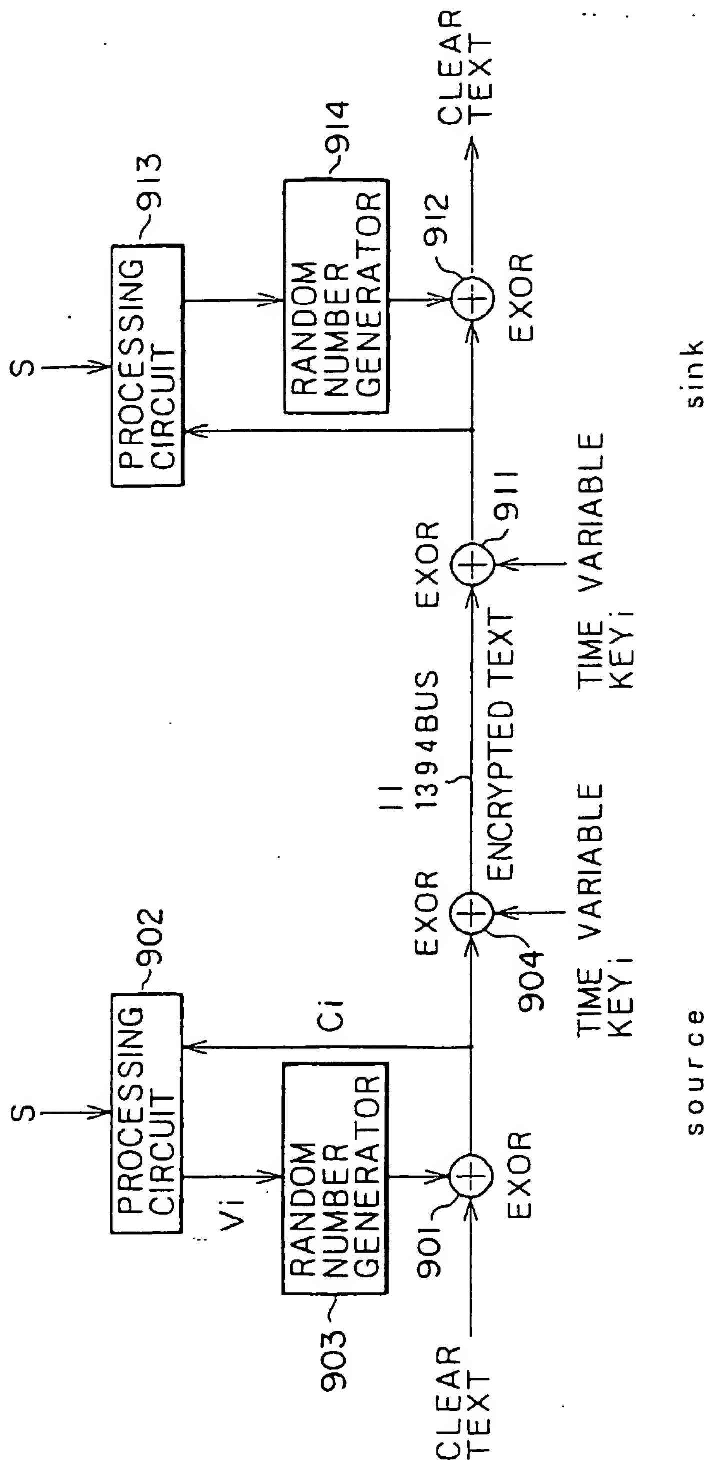

CLEAR TEXT
903
RANDOM NUMBER GENERATOR
901
EXOR
Vi
S
PROCESSING CIRCUIT
902
Ci
EXOR
904
TIME VARIABLE KEYi
source
11
1394BUS
ENCRYPTED TEXT
EXOR
911
TIME VARIABLE KEYi
S
PROCESSING CIRCUIT
913
RANDOM NUMBER GENERATOR
914
912
EXOR
CLEAR TEXT
sink

FIG. 29

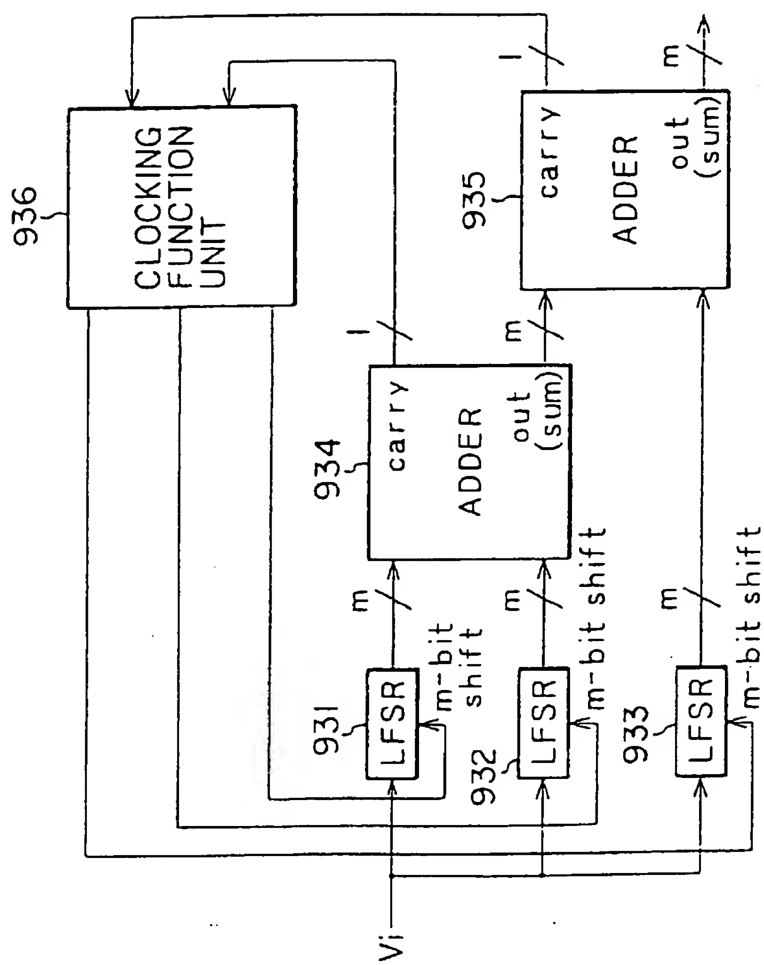

936
CLOCKING FUNCTION UNIT
l
935
carry
ADDER
out (sum)
m
l
934
carry
ADDER
out (sum)
m
931
LFSR
m-bit shift
932
LFSR
m-bit shift
933
LFSR
m-bit shift
Vi

FIG. 30

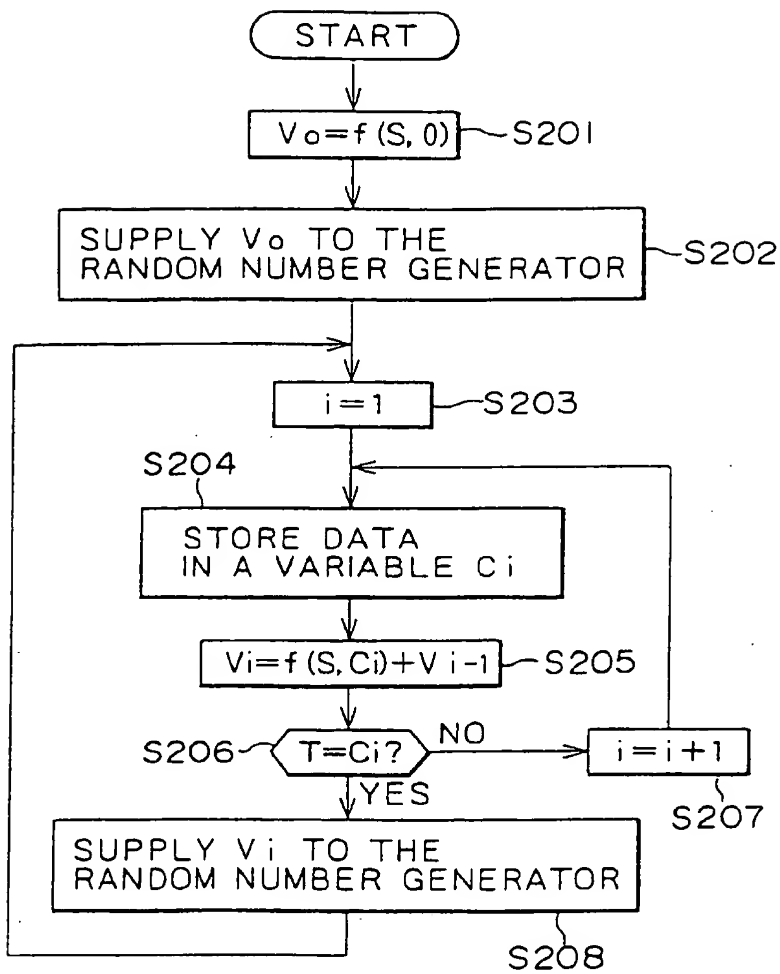

START
Vo=f (S, 0)
S201
SUPPLY Vo TO THE RANDOM NUMBER GENERATOR
S202
i=1
S203
S204
STORE DATA IN A VARIABLE Ci
Vi=f (S, Ci)+V i-1
S205
S206
T=Ci?
NO
i=i+1
S207
YES
SUPPLY Vi TO THE RANDOM NUMBER GENERATOR
S208

FIG. 31

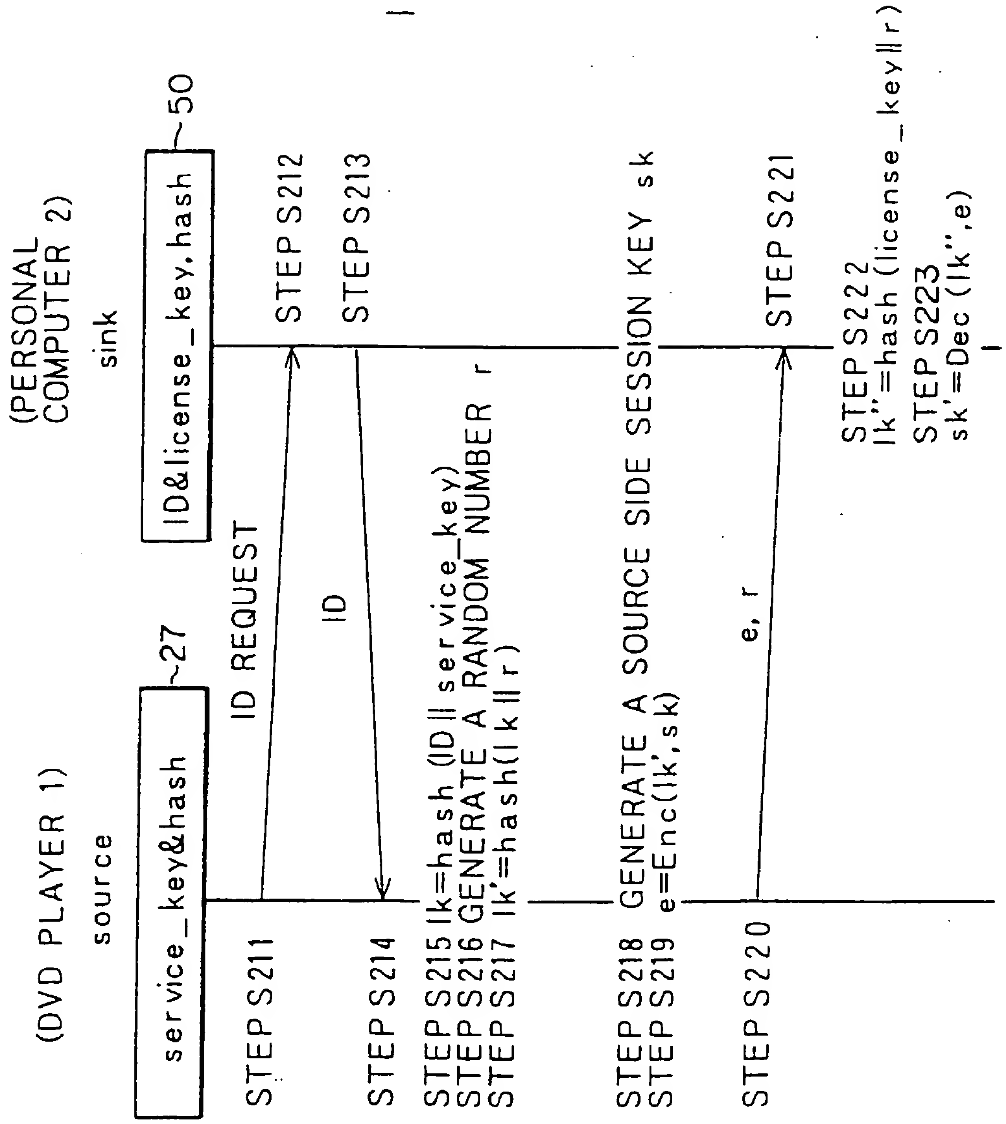
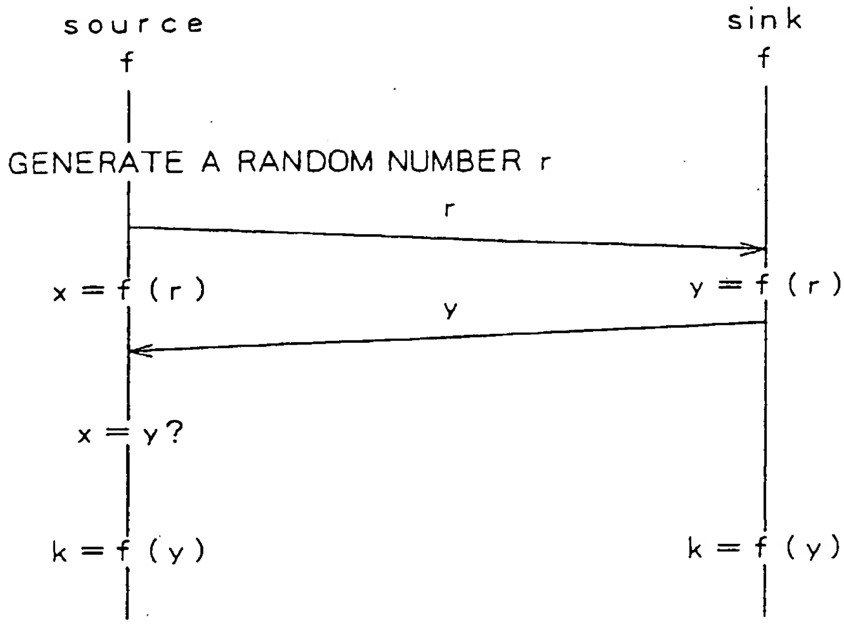

(DVD PLAYER 1)
source
service_key&hash
27
(PERSONAL COMPUTER 2)
sink
ID&license_key,hash
50
STEP S211
ID REQUEST
STEP S212
STEP S213
ID
STEP S214
STEP S215 lk=hash (ID || service_key)
STEP S216 GENERATE A RANDOM NUMBER r
STEP S217 lk'=hash(lk || r)
STEP S218 GENERATE A SOURCE SIDE SESSION KEY sk
STEP S219 e=Enc(lk',sk)
STEP S220
e, r
STEP S221
STEP S222
lk''=hash (license_key || r)
STEP S223
sk'=Dec (lk'',e)

FIG. 32


source
f
sink
f
GENERATE A RANDOM NUMBER r
r
x = f (r)
y = f (r)
y
x = y?
k = f (y)
k = f (y)